An Effective Buffer Management Policy for Opportunistic Networks

Yin Chen[1], Wenbin Yao[1(✉)], Ming Zong[2], and Dongbin Wang[3]

[1] Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia,
School of Computer Science, Beijing University of Posts and Telecommunications,
NO. 10, Xitucheng Road, Beijing 100876, China
chenyin0629@163.com, yaowenbin_cdc@163.com
[2] Yingcai Honors College, University of Electronic Science and Technology of China,
Chengdu 610054, China
[3] National Engineering Laboratory for Mobile Network Security,
Beijing University of Posts and Telecommunications,
NO. 10, Xitucheng Road, Beijing 100876, China

Abstract. Opportunistic networks are wireless networks where disruptions may occur frequently due to the challenging environments. Multiple message replicas have to be propagated to improve delivery probability; combining long-term storage with replication gives rise to a high storage overhead. Many forward/drop policies have been proposed to achieve high delivery ratio, low latencies and low overheads. These policies have improved the performance of opportunistic networks to some extent. However, they all have their own disadvantages. Therefore, an efficient buffer management policy based on the average encounter frequency and the average encounter duration of nodes is proposed in this paper. Simulation results show that our buffer management policy has better performance than the existing DO, DF, MDC-SR and the ACF-based policy.

Keywords: Opportunistic networks · Forward policy · Drop policy · Average encounter frequency · Average encounter duration

1 Introduction

Opportunistic Networks [1] are networks that utilize opportunistic connectivity and node mobility to relay and carry messages around, respectively. This approach is applied in sparse sensor networks for wildlife tracking and habitat monitoring [2], deep-space interplanetary networks [3], vehicle ad hoc networks [4] and military networks [5], where assume that a complete end-to-end path between any pair of nodes may not exist all the time due to sparse coverage, malicious attacks, etc.. In order to increase the probability of delivery, the existing routing mechanisms often replicate the bundles many times as designed in Epidemic routing [6], where two nodes will always exchange all their non-common messages when encounters with each other. At the same time, long-term storage combining with the message replication imposes a high storage overhead on nodes [7]. A large amount of effort has been invested in the design of efficient

S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 242–251, 2017.
DOI: 10.1007/978-3-319-59288-6_22

routing algorithms for opportunistic networks till now. However, there has not been equivalent focus on message scheduling and dropping policies. It has been proved in [8, 9] that the buffer constraints and management policies have severe influence upon the relative and absolute performance of opportunistic networks routing.

In this paper, we propose a novel buffer management policy based on data transmission probability, which is derived from the average encounter frequency and the average encounter duration. Particularly, we also take the size and the time to live of bundles into consideration when determining which bundles to forward and which ones to drop.

The rest of this paper is organized as follows: Sect. 2 describes the current state of the art of drop/forward policies. In Sect. 3, we present the forwarding and dropping policy proposed in this paper. Then, we evaluate our method in epidemic routing by Opportunistic Network Environment (ONE) simulator. Finally, a conclusion about our work is made in Sect. 5.

2 Related Work

In this section, we describe various drop/forward policies introduced lately, most of which are mainly applied in delay tolerant networks or opportunistic networks. Up to now, a sequence of simple drop schemes have been proposed, such as Drop Random (DR), Drop Oldest (DO), Drop Youngest (DY), Drop Front (DF) and Drop Last (DL). In DR, a node simply drops a bundle at random. DO and DY discard bundles based on time-to-live (TTL). DO drops the bundle with shortest TTL; it assumes that a short TTL indicates the bundle has been in the network for a long time, and therefore, is more likely to have been delivered. In contrast, DY drops the bundle with longest remaining life time in the first place. DF considers the arrival time of a bundle and handles the queue in FIFO order; the bundle first entered into the queue will be dropped when the buffer is full. Similar to DF, Drop Last (a.k.a. Drop Tail) simply removes the newly received bundle. In [8], Zhang et al. present an analysis of the impacts of buffer constraints and short contact duration when using epidemic routing protocol and evaluated some of the said buffer management policies. The conclusion is that DF outperforms DL in terms of both delivery rate and delivery delay of bundles. But these simple buffer management policies previously described just take a single factor into account or even provide no mechanism for preferential forwarding and dropping, which are actually affected by several factors. Thereby they are not appropriate and exact for the optimal selection of bundle from buffer overflow.

In [10], Lindgren and Phanse evaluate the following dropping and scheduling policies under the Prophet routing protocol: first-in-first-out (FIFO), most forwarded first (MOFO), most favorably forwarded first (MOPR), shortest life time first (SHLI) and least probable first (LEPR). It demonstrates that MOFO policy presents the best performance among all these diverse drop policies in terms of all different buffer sizes. However, MOFO policy does not consider the lifetime of bundles; in this case, a bundle with insufficient lifetime for delivery but has not been forwarded most will not be dropped. Besides, a large number of dropped bundles may be forwarded again to the same node in the future due to the mobility mode, which is quite a waste of network resource. A buffer management policy called Message Drop Control Source Relay (MDC-SR) [11] that controls the number of dropped bundles is proposed as a variant of MOFO.

Several drop policies with respect to the size of the stored bundles are proposed in [12–14]. In [12], they drop bundles which have a size equal to or greater than that of the incoming bundle. Similarly, Rashid et al. in [13] propose a drop policy that drops the buffered bundle whose size is no less than the mean value. In another work [14], a bundle with the largest size will be dropped simply when congestion occurs. Ayub and Rashid [15] propose a policy, T-drop, where a bundle is dropped if its size is within a certain threshold range. Synthetically, the common problem of the above-mentioned drop policies is that they all ignore how bundle scheduling priorities influence the performance of opportunistic networks. In [16], a novel buffer scheduling and dropping scheme called ACF-based policy is proposed, which considers the average contact frequency among nodes. Yet, the impacts of short contact duration and limited bandwidth on delivery ratio and average delay are disregarded.

In a word, the opportunistic networks are the environment where the partitioned networks and dynamic topologies problems make it difficult to get network-wide information. So local knowledge-based policies have more advantages than the global knowledge-based ones [11]. The DTP-based policy proposed in this paper is based on local knowledge, and has been proved to be an effective solution to the said problem, since it uses the average encounter frequency and the average encounter duration collectively to decide which bundle to forward and which one to drop.

3 Approach

N nodes with finite buffer are assumed to be in the network; each of them has a unique identity number ranging from 1 to N. Nodes move independently of each other at different speed. Source nodes periodically generate random-size bundles, which must be delivered to their destinations within a given TTL. Moreover, each node records its encounter times and duration with other nodes. Nodes update these records when they meet each other. Obviously, if a node has met other nodes many times in the past, it is more likely to encounter with any of them again in the near future. With the contact frequency between two nodes, each node can estimate the possibility for a bundle to reach the destination. However, the times of nodes' encounter cannot reflect the real ability of communication anymore. In addition, contact duration is one of the determining factors of data transmission capability. Therefore, basing on average encounter duration and average encounter frequency, we propose a data transmission probability function, which is used to determine the forwarding priority of a bundle at each contact, and to determine which bundles to drop when congestion occurs. Formally, we define the average encounter frequency and average encounter duration time as follows:

The average encounter frequency (AEF) is defined as the number of encounters between two nodes in the unit time. And the average encounter duration (AED) is defined as the average duration time of several encounters between two nodes in constant time. Let $c_t(m, n)$ describe whether node m and node n encounter at time t. If node m encounters with node n, $c_t(m, n) = 1$; if not, $c_t(m, n) = 0$. Thus, $n_T(m, n)$, the total number of encounters between node m and node n in the given time interval T, can be calculated as follows:

$$n_T(m, n) = n_{t_0+T}(m, n) - n_{t_0}(m, n) = \sum_{t_0}^{t_0+T} c_t(m, n), \tag{1}$$

$$AEF_n^m = f_T(m, n) = \frac{n_T(m, n)}{T} = \frac{\sum_{t_0}^{t_0+T} c_t(m, n)}{T}. \tag{2}$$

Specifically, in Eq. (1), t_0 is an initial time of the networks. According to Eq. (1), the variable AEF, which represents a node's encounter frequency with others, can be computed with Eq. (2). If node m and node n encounter each other at time t, let $d_t(m, n)$ be the duration time of this encounter. Consequently, the variable $s_T(m, n)$ which denotes the total duration time of encounters between node m and node n in the given time interval T can be computed as follows:

$$s_T(m, n) = s_{t_0+T}(m, n) - s_{t_0}(m, n) = \sum_{t_0}^{t_0+T} d_t(m, n), \tag{3}$$

$$AED_n^m = t_T(m, n) = \frac{s_T(m, n)}{n_T(m, n)} = \frac{\sum_{t_0}^{t_0+T} d_t(m, n)}{\sum_{t_0}^{t_0+T} c_t(m, n)}. \tag{4}$$

The meaning of t_0 is same as that in Eq. (1). Combining Eq. (2) with Eq. (3), we can derive approximate calculation of the variable AED, which denotes the average duration time of encounters in the interval T as shown in Eq. (4). Then, for each interval T, data transmission probability (DTP), which indicates the capability to deliver bundles to their destinations through encounters, is decided by average encounter frequency and average encounter duration. Specifically,

$$DTP_n^m = p_T(m, n) = \alpha f_T(m, n) + \beta t_T(m, n), \tag{5}$$

where α and β are normalized parameters of $f_T(m, n)$ and $t_T(m, n)$ respectively.

3.1 Forward Policy

In addition to DTP described above, the size and the time to live (TTL) of the bundles also have effects on whether bundles can be transferred to destination successfully. With a limited bandwidth and short contact duration, nodes are not able to forward all bundles in their buffer to other nodes. In general, bundles with small size are more likely to be delivered successfully. A long TTL implies that a bundle still has a long time to propagate in the network, which may be helpful to deliver the bundle to its destination.

In order to introduce forward strategy, we will first consider a contact between node a and b. Assume that node a plays a role as a sender, whilst node b is a receiver. We will show the details of forward policy based on DTP in Algorithm 1.

Algorithm 1. forward policy

```
while connection between node a and b is up
  do $SV$s are exchanged between node a and b
     node a pushes bundles not in b's buffer into forwardSelection
     for every bundle $i$ in forwardSelection
         do if $d_i = b$
             then push bundle $i$ into directForwardQueue
                 if $S_m = S_n$
                     then sort(TTL, directForwardQueue, desc)
                 else sort($S_i$, directForwardQueue, asc)
                 end if
           else if $DTP^{b}_{d_i} > DTP^{a}_{d_i}$
               then push bundle $i$ into forwardQueue
                   if $DTP^{b}_{d_m} = DTP^{b}_{d_n}$ and $S_m = S_n$
                       then sort(TTL, forwardQueue, desc)
                   else if $DTP^{b}_{d_m} = DTP^{b}_{d_n}$ and $S_m \neq S_n$
                        then sort($S_i$, forwardQueue, asc)
                   else sort($DTP^{b}_{d_i}$, forwardQueue, desc)
                   end if
           end if
     end for
     forward(directForwardQueue, b)
     forward(forwardQueue, b)
end while
```

In Algorithm 1, *SV* is a summary vector of bundles at a node. While a connection between node a and b is up, *SV*s are exchanged each other. Then node a pushes bundles not in b's buffer into *forwardSelection* which is a set of bundles to be forwarded. d_i and S_i denote the destination and the size of the *ith* bundle at node a, respectively. Similarly, both *directForwardQueue* and *forwardQueue* are queues of bundles to be forwarded. If bundle *i* in *forwardSelection* is destined to node b, it will be pushed into *directForwardQueue* directly. Otherwise, if *DTPb di > DTPa di*, it will be pushed into *forwardQueue*. *sort(x, y, z)* is a custom ranking function which sorts bundles in the specified way. For example, *sort(TTL, directForwardQueue, desc)* means that the bundles in *directForwardQueue* will be sorted in *desc* order of their TTLs. As described in Algorithm 1, the bundles will be sorted by various kinds of methods according to different conditions that they satisfy. Finally, the function *forward (directForwardQueue, b)* and *forward (forwardQueue, b)* are to replicate bundles stored in *directForwardQueue* and *forwardQueue* to node b respectively.

3.2 Drop Policy

A drop policy defines which bundle to drop if the buffer of a node is full and a new bundle is to be accommodated. Due to the essential impacts of the average encounter frequency and the average encounter duration on data transmission capability, we should take DTP into account when determining which bundle to drop. Besides, the size and TTL of bundles are also auxiliary determinants of drop policy. This is because dropping bundles with large size can result in less bundle drop. In addition, a bundle with a short TTL implies it has been in the network for a long time and thus is more likely to have been delivered. The details of drop policy based on DTP are shown in Algorithm 2.

Algorithm 2. drop policy

```
bundle M is forwarded to node a
node a is congested
  for every bundle i in a's buffer
      do if $DTP^a_{dM} > DTP^a_{di}$
          then push bundle i into dropQueue
        end if
  end for
  while dropQueue $\neq$ NULL and $FS_a < S_M$
      do for bundle m and n in dropQueue
          do if $DTP^a_{dm} = DTP^a_{dn}$ and $S_m = S_n$
              then sort(TTL, dropQueue, asc)
            else if $DTP^a_{dm} = DTP^a_{dn}$ and $S_m \neq S_n$
                then sort($S_i$, dropQueue, desc)
            else sort($DTP^a_{di}$, dropQueue, asc)
            end if
         end for
        drop(dropQueue)
  end while
  if $FS_a \geq S_M$
      then store bundle M
  else refuse to store bundle M
  end if
```

When a new bundle M is forwarded to node a, it becomes congested and runs Algorithm 2. If *DTPa dM > DTPa di*, bundle *i* will be pushed into *dropQueue* which is a queue of bundles to be dropped. The meaning of d_i, S_i, *sort(x, y, z)* are same as those of Algorithm 1. Additionally, FS_a denotes the free space of node a's buffer. The bundles will be arranged according to the TTL, S_i or *DTP* values of bundles. Then *drop (dropQueue)* is to drop the bundles in *dropQueue* sequentially until *dropQueue* becomes empty or node a has enough space for bundle M. If $FS_a \geq S_M$, node a will store bundle M in its buffer; otherwise, it refuses to store bundle M.

4 Simulation and Results

In this section, we will evaluate the DTP-based policy proposed in this paper against the following local knowledge policies: DO, DF, MDC-SR and the ACF-based policy in Epidemic routing with ONE simulator [17]. The main simulation parameters used in the experiment are listed in Table 1.

Table 1. Simulation parameters.

Simulation area	4500*3400 m^2
Number of nodes	100
Movement model	Random Way Point
Speed of nodes	Randomly selected in (0,15)m/s
Size of messages	Randomly selected in [50 KB,100 KB]
Message generation interval	25 s,35 s
Transmission range	10 m
Transmission speed	250KBps
Simulation duration	12 h

The definitions of the main performance metrics are defined as follows:

(1) Delivery ratio: the ratio of the number of bundles received by destination nodes to the number of bundles created by source nodes, shown in Eq. (6).

$$delivery\_ratio = \frac{N_d}{N_c} \times 100\%. \tag{6}$$

The terms N_d and N_c respectively represent the number of bundles delivered to destination nodes and the number of bundles created by source nodes.

(2) Average delay: the average latency of all bundles received by destination nodes, shown in Eq. (7).

$$average\_delay = \frac{\sum_{i=1}^{N_d} D_i}{N_d}. \tag{7}$$

D_i denotes the delay of the *ith* bundle received by destination nodes.

(3) Overhead: the ratio of the number of bundles received by destination nodes to the number of bundles carried by them, shown in Eq. (8).

$$overhead = \frac{N_d}{N_{ca}} \times 100\%. \tag{8}$$

N_{ca} is the number of bundles carried by destination nodes.

Firstly, Fig. 1 reveals the comparison of DO, DF, MDC-SR, the ACF-Based and the DTP-Based policy proposed in this paper with respect to delivery ratio. As a whole, the delivery ratio of all policies increase rapidly at early stage and then attain a stable state. The impact of buffer size on delivery ratio decreases gradually with the increment of buffer size. When the buffer of node is enough to accommodate all bundles, buffer size is no longer regarded as the bottleneck of delivery ratio increase. The DTP-Based policy has up to 7.4% improvement as compared with DO and up to 23.2% improvement as compared with the ACF-Based policy. As shown in this graph, the ACF-Based policy and MDC-SR show lower delivery ratio comparing with others. This is because they just consider delivery probability of bundles based on the encounter frequency of nodes, which is inexact in random way point mobility model. However, the DTP-Based policy also takes the encounter duration, the size and the time to live of nodes into account when estimating how likely a bundle will be delivered.

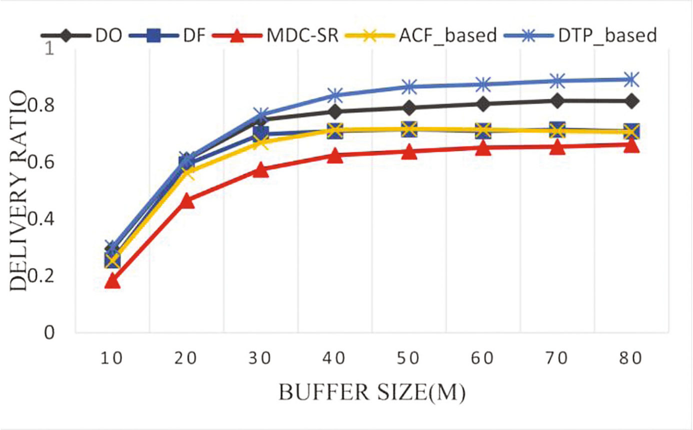

Fig. 1. Delivery ratio under random way point model with different node buffer sizes.

Figure 2 shows the average delay of DO, DF, MDC-SR, the ACF-Based and the DTP-Based policy. The average delay of MDC-SR and the ACF-Based policy are larger than that of DO and DF. In random way point mobility model, it may be inaccurate to use nodes' encounter frequency to estimate delivery delay of bundles. But the DTP-Based policy also takes advantage of node's encounter duration, bundle lifetime and bundle size to address the said issue. Consequently, comparing with MDC-SR and the ACF-Based policy, it has up to 40.6% and 22.6% reduction in average delay, respectively. Furthermore, the average delay of the DTP-Based policy keeps more stable than others with the increasing of buffer size.

Finally, it can be seen that the overhead of the DTP-based policy is much lower than that of others, especially than that of MDC-SR, when they are in the same scenario. The bundles are delivered to the nodes with a higher DTP value, which indicates the probability of meeting the destination nodes. Comparing with MDC-SR and the ACF-based policy, the DTP-based policy takes both the average encounter frequency and the average encounter duration into consideration. Apart from this, the size and the time to live of bundles are also auxiliary determinants of the DTP-based policy. Therefore, our policy has achieved a much better and more stable performance (Fig. 3).

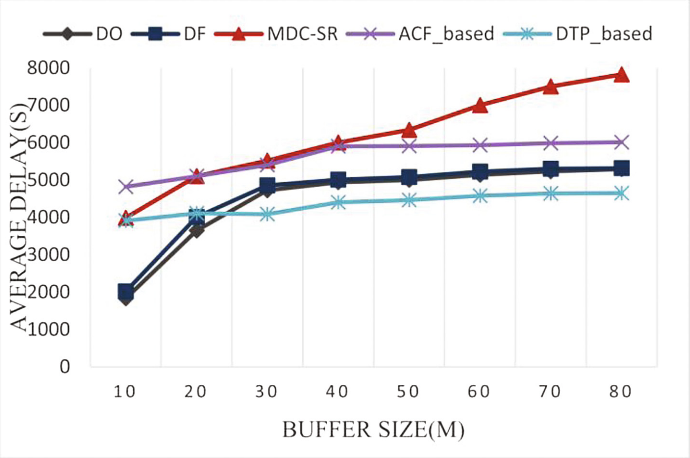

Fig. 2. Average delay under random way point model with different node buffer sizes.

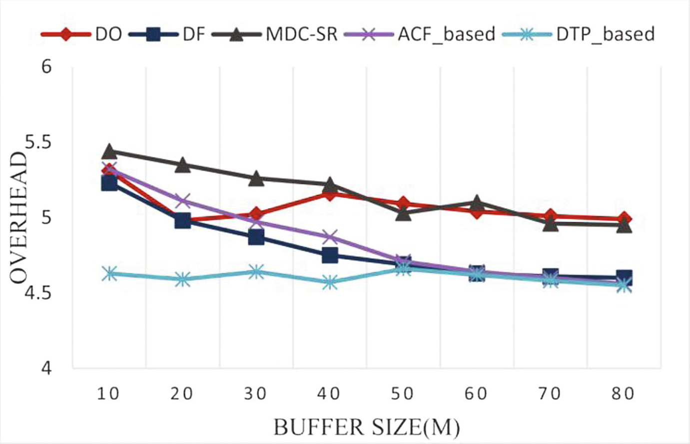

Fig. 3. Overhead under random way point model with different node buffer sizes.

5 Conclusion

In this paper, we investigate the problems of buffer management policies in the opportunistic networks. We propose a new metric derived by the average encounter frequency and the average encounter duration, called as data transmission probability, indicates the capability of node to deliver bundle to its destination through encounter.

Then an effective scheduling and dropping policy based on the DTP, TTL value and the size of bundles is proposed to decide which bundle to forward and which bundle to drop. The simulation results show that our strategy outperforms DO, DF, MDC-SR and the ACF-based policy in terms of delivery ratio, average delay and overhead.

Acknowledgments. This work was partly supported by the NSFC-Guangdong Joint Found (U1501254) and the Co-construction Program with the Beijing Municipal Commission of Education and the Ministry of Science and Technology of China(2012BAH45B01) and the Fundamental Research Funds for the Central Universities (BUPT2011RCZJ16, 2014ZD03-03) and China Information Security Special Fund (NDRC).

References

1. Xiong, Y.P., Sun, L.M., Niu, J.W., et al.: Opportunistic networks. J. Softw. **20**, 124–137 (2009)
2. Tekdas, O., Bhadauria, D., Isler, V.: Efficient data collection from wireless nodes under the two-ring communication model. Int. J. Robot. Res. **31**, 774–784 (2012)
3. Wang C., Wang W., Sohraby K., et al.: QoS optimized and energy efficient power control for deep space multimedia communications in InterPlaNetary networks. In: International Conference on Wireless for Space and Extreme Environments, pp. 1–3. IEEE (2013)
4. Burgess, J., Gallagher, B., Jensen, D., et al.: MaxProp: Routing for vehicle-based disruption-tolerant networks. In: Proceedings - IEEE INFOCOM, pp. 1–11 (2006)
5. Krishnan, R., Basu, P., Mikkelson, J.M., et al.: The SPINDLE disruption-tolerant networking system. In: Military Communications Conference, pp. 1017–1023. IEEE (2007)
6. Rashid, S., Ayub, Q., Zahid, M.S.M., et al.: Impact of mobility models on DLA (Drop Largest) optimized DTN epidemic routing protocol. Int. J. Comput. Appl. **18**, 35–39 (2011)
7. Jain, S., Chawla, M.: Survey of buffer management policies for delay tolerant networks. J. Eng. (2014)
8. Zhang, X., Neglia, G., Kurose, J., et al.: Performance modeling of epidemic routing. Comput. Netw. **51**, 827–839 (2006)
9. Krifa, A., Barakat, C., Spyropoulos, T.: An optimal joint scheduling and drop policy for delay tolerant networks. In: World of Wireless, Mobile and Multimedia Networks, pp. 1–6. IEEE, California (2008)
10. Lindgren, A., Phanse, K.S.: Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks. In: International Conference on Communication System Software and Middleware, pp. 1–10. IEEE (2006)
11. Rashid, S., Ayub, Q., Zahid, M.S.M., et al.: Message drop control buffer management policy for DTN routing protocols. Wireless Pers. Commun. **72**, 653–669 (2013)
12. Rashid, S., Ayub, Q., Soperi, M.Z.M., et al.: E DROP an effective drop buffer management policy for DTN routing protocols. Int. J. Comput. Appl. **13**, 8–13 (2010)
13. Rashid, S., Abdullah, A.H., Soperi, M., et al.: Mean drop an effectual buffer management policy for delay tolerant network (2012)
14. Ayub, Q., Rashid, S., Zahid, M.S.M.: Buffer scheduling policy for opportunitic networks (2013)
15. Ayub, Q., Rashid, S.: T-Drop: An optimal buffer management policy to improve QOS in DTN routing protocols. J. Comput. **2**, 46–50 (2010)
16. Tang, L., Chai, Y., Li, Y., et al.: Buffer management policies in opportunistic networks. J. Comput. Inf. Syst. **8**, 5149–5159 (2012)
17. Keränen, A., Ott, J., et al.: The ONE simulator for DTN protocol evaluation. In: International Conference on Simulation TOOLS and Techniques for Communications, Networks and Systems, Simutools, Rome, Italy (2009)