Survey of buffer management policies for delay tolerant networks

Sweta Jain[1], *Meenu Chawla*[1]

Department of CSE and IT, Maulana Azad National Institute of Technology, Bhopal, India
E-mail: chawlam@manit.ac.in

Published in *The Journal of Engineering*; Received on 5th March 2014; Accepted on 5th March 2014

Abstract: Delay tolerant networks (DTN) are a class of networks that are a subset of the traditional mobile *ad-hoc* networks. It differs from mobile *ad hoc* networks (MANETs) in the sense that it can withstand high delays in delivering data because of frequent network partitions, limited bandwidth and storage constraints persisting in such networks. Owing to these inherent characteristics of the delay tolerant networks improving delivery ratio in such networks depends on two main factors-use of routing strategy and a good buffer management policy. Many routing protocols have been proposed in the literature for DTN. Buffer management is a very important factor in DTN because of the very limited buffer space available in DTN nodes. Although a scheduling policy in DTN determines which message has to be forwarded first, the dropping policy decides which messages are to be dropped in case of buffer overflow. This Letter presents a survey of the existing buffer management policies proposed for DTN and discusses the pros and cons of these approaches. The buffer management techniques have been classified on the basis of information used by them whether they are based on local information of messages available at the node or global information of all the messages in the network.

1 Introduction

Millions of devices are connected to each other with the help of Internet operating on the transfer control protocol/internet protocol (TCP/IP) to provide a reliable communication through a standard set of protocols. The usability of Internet however relies on some important assumptions like continuous bidirectional end-to-end path, short round trips, symmetric data rates, low error rates etc. However with the proliferation of mobile devices in each and every aspect of the current world, the Internet has limitations to provide a satisfactory performance in connecting the millions of mobile devices popping up. In networks like terrestrial networks, military networks etc. where frequent network partitions are likely to occur because of the frequent link disruptions caused by obstacles or power shut downs using a technology like TCP/IP or traditional MANET routing protocols, which require an end-to-end connection, cannot ensure a successful delivery.

The environments that are characterised by intermittent connectivity, long or variable delays, asymmetric data rates and high error rates are known as delay tolerant networks [1–3]. Since the chances for a destination to come into contact with a source node are very less in a DTN, these networks employ a store and forward paradigm to forward the messages to the destination. This means that whenever a node has a message for another node and this node is not in its vicinity, then the node will forward the message to the next encountered node. The encountered node stores this message in its buffer in order to forward it to the destination.

The store and forward approaches have two major problems that have evolved two separate research areas in the study of DTNs. The first problem is that forwarding messages to any encountered node is not feasible. The main reason behind this is that forwarding to a large number of nodes puts a burden on limited node resources like buffer, energy and also wastes limited network bandwidth. This also leads to network contention because of the flooding of messages. Thus, nodes to which message has to be forwarded should be chosen carefully. This problem is addressed in the design of routing protocol. Another problem arises because of limited buffer space available at DTN nodes. Since DTN nodes work on store carry and forward mechanism their buffer space gets filled up quickly. A situation can arise where each node buffer is full with messages from other nodes in the network. In such a situation, when a new encounter happens, either the node has to drop a message from its buffer or it may have to deny the sending node of a new message transfer. This problem is taken care by the buffer management policies in a DTN.

Buffer management is very important in DTN because the combination of long-term storage of messages and the message replication places a high bandwidth and storage overhead on nodes [4]. When the buffer is full, often in order to accommodate a new message a DTN node will have to drop an important message. If an efficient dropping policy is implemented that can prioritise message drop sequence, it will have a huge impact on the delivery ratio of the network. Not only dropping policies define buffer management in DTN, but also the scheduling policies. Since DTN suffers from partial transmissions, sorting the messages in the order of their relevance is very important for their successful delivery. Partial transmissions occur when two nodes are transmitting the messages and their transmission has to be aborted because of some link failure or power shut down by either of the nodes. Thus, both scheduling and dropping policies play a huge role in the delivery performance of a DTN network.

The goal of an efficient buffer management policy can be summarised as below. The primary and necessary goal must be to improve the delivery ratio of the network. Even though the DTNs are made to tolerate delays minimising the average delay of delivering messages is another factor to be taken into consideration while developing an efficient buffer management policy. If delay is too high, the messages can reach the destination when almost the relevance of the message is lost. To preserve the resource consumption, a buffer management policy must also try to reduce the overhead ratio, that is, minimises the number of relays to which a message is forwarded.

In this paper, various DTN buffer management policies that have been introduced in the literature are surveyed and reviewed. Buffer management policies have been mainly classified on the basis whether they use locally available information or globally available information about all messages. Each algorithm is described in brief along with their advantages and disadvantages with respect to a DTN environment. The rest of this paper is organised as follows. Section 2 introduces the various buffer management schemes and Section 3 summarises the various buffer management policies. Section 4 provides the conclusion of this paper.

2 Buffer management policies in DTN

In this section, we describe the various buffer management policies for the delay tolerant networks that have appeared in the literature. The methods are described in the order in which they were introduced in the past years along with their shortcomings and advantages.

2.1 Drop-random, drop-least-recently received, drop oldest and drop-least-encountered

Davis *et al.* [5] introduced four dropping policies in 2001. The goal of the dropping policies was to increase the delivery performance in highly partitioned *ad-hoc* networks. This paper stressed the importance of these policies in terms of the increasing use of wearable computer for packet transport mechanisms. Four dropping policies were introduced in this paper: 'drop random (DRA), drop-least-recently-received (DLR), drop oldest (DOA) and drop-least-encountered (DLE)'. Among the four strategies used, DRA uses the simplest strategy, that is, when the buffer is full drop randomly selected packets until space for newly arrived packet is created. This strategy does not take into account the network conditions and is simply blind. The DLR drops the packets that have stayed in the agent's buffer for a long time. The idea behind such an approach is that DLR assumes the packets that have stayed for a long time in the buffer must have been forwarded the most number of times in the past encounters and therefore can be dropped. It assumes that such packets have enough replicas in other nodes so that their probability of delivery is higher than other packets in the buffer. The DOA policy drops the messages that have stayed for the longest time in the network. Such a policy requires network information and is therefore more adaptive. The messages that have stayed for a long time in the network have higher probability to be already delivered and therefore are dropped.

The major contribution of this paper however is DLE. In this policy, messages are dropped based on the estimated likelihood of delivery. It takes into consideration agent location and movement for its implementation and hence is more adaptive than the other three approaches. The DLE sorts the packets according to the 'relative ability' of two agents to forward a packet to the destination. To calculate the relative ability, each agent (node) maintains a list containing the agent addresses of other agents in the network. For each time step, agent A updates the meeting time for another agent C, with respect to co-located agent B, using the following rule

$$M_{t+1}(A, C) = \begin{cases} \lambda M_t(A, C), & \text{if} \quad \text{none co} - \text{located} \\ \lambda M_t(A, C) + 1, & \text{if} \quad C = B \\ \lambda M_t(A, C) + \alpha M_t(B, C), & \text{for all} \quad C \neq B \end{cases}$$

where $M_t(A, C)$ is the meeting value at time t, $\alpha = 0.1$ is a parameter that decides how much portion of B's meeting value is to be added with A and $\lambda = 0.95$ is the decay rate of the meeting value. The value $M_t(A, C)$ is initially zero for all node pairs.

The meeting value is updated in three steps: (i) update the meeting value whenever node A meets node B; (ii) update the transitivity in meeting values, for example, if node A encounters node B and node B has a high encounter value for C, then A is a good candidate for passing packets to C also; and (iii) with time the meeting values have to be aged if nodes A and B do not meet for a long time.

To sort the packets in the buffer according to the 'relative ability' of two nodes to deliver the packet to the destination, whenever node A meets B the packets are sorted according to $M_t(A, C)$–$M_t(B, C)$, where C is the destination of the message. Such a strategy ensures that the packets can always move from agents that have less probability to deliver the packets to the destination to agents that have a high delivery probability with the destination. This proves that DLE works best when compared with other algorithms because of its adaptiveness with the network conditions.

2.2 Buffer management in MaxProp

MaxProp [6] is one of the earliest protocols that combined buffer management along with a routing strategy. Routing protocols proposed earlier to it like 'Epidemic' [7], 'Prophet' [8], 'Spray and Wait' [9] did not specify any specific buffer management technique. For effectively managing the buffer, the buffer is logically divided into two parts based on whether the hop count of the packets have a hop count less than a threshold t hops. If the hop count is below t counts, the packets are then sorted in the increasing order of the hop count. For those packets which are having a hop count greater than t then they are sorted in the decreasing order of 'delivery likelihood'. Delivery likelihood is a new metric used in MaxProp to prioritise the packets based on the cost that it takes to reach the destination.

When a node encounter occurs, packets which have hop count less than threshold t are forwarded first. If all such packets are forwarded then only the packets having hop count greater than threshold t are given a chance for transmission. By giving priority to the first buffer section, the MaxProp is giving importance to those packets whose hop count is very less. The assumption is that the packets with a very small hop count are relatively new in the network and have not traversed far in the network. Therefore such packets are given more importance. For packets above the threshold t, they are sorted in decreasing order of 'delivery likelihood'. Those packets with a high value for this metric are forwarded first. Thus, the MaxProp uses an efficient scheduling policy for forwarding the packets to the relay nodes to ensure better packet delivery. For a dropping policy, when a packet is to be dropped it is dropped from the tail of the buffer. The tail of the buffer contains the packet with the lowest 'delivery likelihood' and is not a new packet since it is in the second section of the buffer. Such a packet can be easily dropped since it does not have enough priority. MaxProp shows a high delivery ratio often in comparison with the other basic routing protocols in DTN thereby emphasising the importance of buffer management. The communication cost for MaxProp is however too high because of the calculation of shortest path to the destination for estimating the 'delivery likelihood'.

2.3 Evaluation of queuing and scheduling policies in DTN

Lindgren and Phanse proposed various dropping and scheduling policies in [10]. These policies were based on the 'delivery probability' concept in Prophet routing protocol [8]. The authors proved that considering delivery predictability while taking forwarding and dropping decision is far better than using a simple random pruning as used in epidemic routing. The dropping policies compared here are: first-in-first-out (FIFO), most forwarded first (MOFO), most favourably forwarded first (MOPR), shortest life first (SHLI) and least probable first (LEPR).

Among the dropping policies, FIFO is the simplest policy. The packets in the queue are sorted for dropping in the order those messages that were received earlier. The message that came to the queue first will be dropped, then the second and so on.

MOFO forwards the messages that have been forwarded most number of times. For this the node keeps a counter each time the messages are forwarded. When a packet is to be dropped because of buffer overflow, the one which was forwarded the most is deleted. This is because such packets have already reached many nodes and have high probability to get delivered and can be easily dropped.

MOPR uses a metric FP to count the most favourably forwarded message. The value of FP is initialised to zero and each time it is forwarded the value is updated using the equation $\text{FP} = \text{FP}_{\text{old}} + P$, 'here' P denotes the delivery probability value with which the

message was forwarded. Although dropping the message with highest FP value is dropped first.

SHLI is based on the concept of message time-to-live (TTL). The message having lowest remaining TTL is dropped first. This is based on the idea that a message having a low remaining TTL does not have enough time to reach the destination and will get expired soon and therefore it can be easily dropped.

LEPR drops the message for which it has the lowest P-value. However, if the source has a small delivery probability for the destination of a message, this message will never get forwarded and will not be spread in the network further reducing its chance of reaching the destination.

The authors proved that MOFO is best in terms of delivery ratio and SHLI for average delay. This is because MOFO ensures that each packet be forwarded at least once before it is dropped. In case of SHLI, the packets that were never forwarded can also be dropped.

The forwarding strategy of Prophet was slightly modified in each scheduling mechanism proposed in [8].

In GRTR, the forwarding strategy is same as that of the Prophet. The message with destination D is forwarded to an encountered node B by a current node A only if $P(B, D) > P(A, D)$, where $P(i, j)$ is the delivery probability between two nodes i and j.

The next scheduling policy GRTRSort selects the messages in the descending order of $P(B, D) - P(A, D)$ and then forwards if $P(B, D) > P(A, D)$. In GRTR, the messages were scanned in a linear way; however, GRTRSort first selects those messages which have a high difference between $P(B, D)$ and $P(A, D)$ and then performs forwarding.

In GRTRMax, messages are sorted in the descending order of $P(B, D)$ and then forwarded if $P(B, D) > P(A, D)$. This is based on the same logic as that of GRTRSort, but it prioritises based on the highest delivery predictability than by maximising the improvements.

The next strategy is not used in Prophet, but in Epidemic and is termed as COIN. Instead of simply forwarding that is used in Epidemic, the method first performs a coin toss to check whether the message is to be forwarded or not. For this a message is forwarded only if a value $X > 0.5$, where $X \in U(0, 1)$.

The simulation results presented in this Letter showed that GRTRSort in combination with MOFO has the highest delivery performance.

2.4 Prioritised epidemic routing

Ram Ramanathan *et al.* proposed a buffer management scheme that considered the topology of the network in [11] known as 'Prioritised Epidemic' routing (PREP). The PREP orders the messages in the node for both transmission and drop by assigning priorities to them. It consists of two mechanisms: (i) a method that estimates the routing cost from a given node to the destination and (ii) a priority mechanism for message processing (deletion and transmission).

The costs for inter-node contacts are computed using a metric known as 'average availability' (AA). It measures the average fraction of time in the near future for which the link will be available for use. When there is a threshold change in the value of AA for a node, then link state advertisements (LSA) with a list of all current links and their associated metrics is generated. The LSA is then circulated through whole network using 'epidemic routing'. The advantage of such dissemination is that when two partitions are joined, the node in each partition gets to learn the entire connected topology. This helps a node i to keep a knowledge of the subgraph from which an LSA was transportable to i during some recent time period. To calculate the routing costs, a cost is assigned to each link using the equation $(1 - \mathrm{AA}) + 0.01$ after which Dijkstra's shortest path algorithm is used to find the lowest cost route.

To implement the priority-based dropping and scheduling policies, each packet is assigned with a drop priority p_d and a transmit priority p_t. For dropping those messages having hop count greater than a threshold V_hc are selected. Then, they are assigned priority according to the cost path to destination from the current node. Those messages that are far from the current node to the destination are dropped first. If the buffer occupancy is still so high that new messages cannot be accommodated, then all the messages have a hop count less than a threshold V_hc are dropped. For transmission, the node calculates the cost to destination of the encountered node. If it has a smaller cost, then the message is places in downstream bin, otherwise in upstream bin. Messages in downstream bin are sorted based on remaining TTL and age of the message.

It can be seen that this method uses the history of contacts for transmission and dropping of the messages. When in traditional flooding, the nodes have no idea of the topology of the peer; this method can obtain an idea of the network to which the message is being forwarded. Thus it is a guided transmission. The disadvantage of this method is the dissemination of LSAs. In a network with high node density, such dissemination leads to high overhead. It is useful in such networks where load is less relative to the network resources.

2.5 Optimal joint scheduling and drop policy for DTNs

Krifa *et al.* proposed an optimal policy for both scheduling and dropping in DTNs in [4]. The approach proposed in global knowledge based scheduling and drop (GBSD) is based on optimising two parameters: (i) maximising delivery ratio and (ii) minimising average delay. The GBSD calculates per-message utility for a given routing metric. To maximise the average delivery rate of all the messages, the message having smallest value of below given utility function is dropped

$$\left(1 - \frac{m_i(T_i)}{L-1}\right)\lambda R_i \exp\left(-\lambda n_i(T_i) R_i\right)$$

where it is assumed that there are K messages that are in the network with L nodes and the elapsed time is T_i for a message i when a drop or forwarding decision is to be made. $m_i(T_i)$ is the number of nodes that have seen the message i since its creation and $n_i(T_i)$ are the nodes that have a copy of the message i. The meeting time between nodes is exponentially distributed by the parameter λ.

The GBSD policy to minimise the average delay drops the message having smallest value of following utility function

$$\frac{1}{n_i(T_i)^2 \lambda}\left(1 - \frac{m_i(T_i)}{L-1}\right)$$

Since the calculation of utility function in GBSD requires complete knowledge about every message in the network, that is, the number of nodes who have seen the copy of message and number of nodes who have a copy of the message, a history based scheduling and drop (HBSD) policy is used to employ a distributed algorithm. This policy learns the network statistically using the network history to obtain an estimate of the global status of the network. With HBSD, the per-message utility for maximising the delivery ratio is

$$\lambda R_i E\left[\left(1 - \frac{M(T)}{L-1}\right)\exp\left(-\lambda R_i N(T)\right)\right]$$

Here, $m_i(T)$ and $n_i(T)$ are supposed to be instances of random variables $M(T)$ and $N(T)$. When global information is not available, the HBSD relies on the method of calculating the average delivery rate using all possible values of $M(T)$ and $N(T)$ and then maximise it. Similarly, the HBSD policy of minimising the average delivery

delay is as follows

$$\frac{E[(L-1-M(T))/N(T)]^2}{\lambda(L-1)\left(L-1-\bar{m}(T)\right)}$$

The HBSD is independent of the actual global state as it is just a function of the locally available history. In [4], the authors mention that for a large number of messages, this policy is expected to lead to the same average delay as in GBSD policy.

Even though this policy provides an optimal framework, it is based on some assumptions like the nodes' movement path is known, the bandwidth is unlimited and all the messages have the same size. Such a situation is very impractical in a DTN and therefore can only be considered as an optimal solution against which other methods can be compared.

2.6 Adaptive optimal buffer management policies (AOBMP) for realistic DTN

Owing to the impractical assumptions made in [4], Li and Qian proposed a new set of optimal policies for buffer management in [12]. This assumes that both the bandwidth and capability for one communication contact is limited and that the messages vary in their size. These match with the real characteristics of a DTN. The system model assumes that there for each node i there are $N_t(t)$ messages in its buffer at time t. The set of messages in i at time t, $PK_i(t)$ is denoted as $\left\{P_i^1,\ P_i^2,\ P_i^3,\ \ldots,\ P_i^{N_i(t)}\right\}$. Each message j in i is denoted as a tuple $\left(\mathrm{Sou}_i^j(t),\ \mathrm{Dst}_i^j(t),\ T_i^j(t),\ s_i^j(t),\ n_i^j(t),\ C_i^j(t)\right)$ which represent source, destination, remaining time, size, the number of copies and transmission opportunity of message j, respectively. The opportunity for packet j to be transmitted to destination Dst_t^j by all the nodes including i is denoted as $C_i^j(t)$. $C_i^j = \left\{(u_1,\ l_1),\ (u_2,\ l_2),\ \ldots,\ \left(u_{n_i^j(t)},\ l_{n_i^j(t)}\right)\right\}$, where the kth tuple represents that there is copy of message j in the buffer of node u_k and in the front of the buffer l_k bytes of other messages are present.

The AOBMP protocol consists of three steps including an 'initialisation, utility computation and message' discarding step. The initialisation step consists of exchanging metadata between two nodes and updating transmission opportunity of messages and parameters of mobility model. In utility computation, the utility function for messages in node i is calculated and the 'message discarding' step calculates a metric if satisfied on which the message shall be discarded.

When the buffer of node i is full at time t_0, the AOBMP policy to maximise the average delivery rate is to discard the message $j_{\max}$ satisfying the following condition

$$j_{\max} = \arg \max_{j\epsilon[1,N_i(t_0)]} \left(\sum_{n=1}^{\infty} \frac{\left(\lambda_2 R_i^j\right)^{n-1}}{(n-1)!} \sum_{k=0}^{n=1} \frac{\left(\lambda_1 T_i^j\right)^k}{k!} \right) \mathrm{e}^{-\lambda_2 R_i^j - \lambda_1 T_i^j}$$

Here, T_i^j and R_i^j are the remaining time and additional time needed to transmit message j, respectively, and λ_1 and λ_2 are the parameters of exponential distribution of inter-meeting time and contact time, respectively.

To minimise the average delay, the node should discard a message which maximises the following function

$$j_{\max} = \arg \max_{j\epsilon[1,N_i(t_0)]} \sum_{n=1}^{\infty} \frac{n}{\lambda_1} \left(1 - \sum_{k=0}^{n} \frac{\left(\lambda_1 T_i^j\right)^k}{k!} \mathrm{e}^{-\lambda_1 T_i^j} \right) \times \frac{\left(\lambda_2 R_i^j\right)^{n-1}}{(n-1)!} \mathrm{e}^{-\lambda_2 R_i^j}$$

Moreover, the AOBMP adapts to each mobility model characteristic by adjusting the mean of exponential distribution with the aid of the information exchanged through the control channel. This approach works well in DTN as the assumptions are more realistic.

2.7 N-drop congestion control strategy under epidemic routing in DTN

Li *et al.* proposed a buffer management strategy specific to Epidemic routing in [13]. In this strategy, the congestion is prevented by designing an efficient dropping policy when the buffer is full. Each node keeps track of the number of times a message has been forwarded since the time of its creation. A threshold N is calculated by each node as a function of its buffer size. As the buffer size increases, the threshold N also increases and can be shown as $N = f$(buffer size). When the buffer is full, the node checks the number of forwarding of each packet in the buffer. The packet whose number of forwarding is greater than or equal to the threshold N is dropped. If all the packets in the buffer have a value less than N, then the packet at the end of the buffer is dropped. This method is proved to be better than the drop-last (DL) packet strategy in which the packets at the tail of the buffer are dropped on the occurrence of congestion. However when all packets have their forwarding count less than N, then the strategy degrades to DL itself. One solution is to rearrange the packets in descending order of their forwarding count and then drop the packet with the highest count. This method moreover does not control the blind forwarding of epidemic which is also another important factor that leads to congestion.

2.8 Performance analysis of scheduling and dropping policies in VDTN

In [14], Vasco *et al.* has shown a performance analysis of various scheduling and dropping policies along with a combination of both methods to show the effect of buffer management in effectively delivering the packets in a vehicular delay tolerant network (VDTN). The scheduling policies considered in this paper were FIFO, random, remaining lifetime (RL) and replicated copies (RCs). The FIFO as known schedules the packets based on their arrival time. The basic concept in FIFO is that the packets that arrived first has to be given the top priority. In random, this idea is challenged because in a typical DTN environment where short encounter opportunities and finite bandwidth is in commonplace giving bundles that arrived first the only chance can lead to starvation of other bundles. Therefore packets are randomly forwarded to avoid the starvation of other packets. This method is shown to have better delivery ratio than FIFO. The next scheduling policy is based on the RL or TTL (RTTL) of the packets. The authors have studied performance of both scheduling policies one in which packets with smallest RTTL are scheduled first [RL ascending order (RL ASC)] and the other method in which the packets with largest RTTL are [RL descending order (RL DSC)] scheduled in the beginning. The idea behind forwarding packets with smallest RTTL is that these packets have small time before their expiry and therefore should be forwarded quickly. However what if such packets gets expired soon before reaching their destinations? So forwarding those packets that can remain until they reach the destination seems good in a DTN where destination nodes may be quite far from the nodes. Another method is scheduling the packets based on the RCs. In this method, each node has to keep track of the number of times it got replicated. Based on this value, we can have two approaches: RC ASC and RC DSC. Either the bundle that has been replicated more is forwarded or the one replicated less is forwarded. These approaches are followed in dropping policies also like FIFO drop, random drop, RL ASC/RL DSC drop and RC ASC/RC DSC drop.

The simulation results in [14] showed that best performances are achieved using a policy that combines both scheduling and

dropping by giving more preferences to the packets with less number of RCs. This analysis proved that in a DTN the best consideration should be given for newly created copies since they have not been diffused much into the network.

2.9 Novel buffer management policy for DTN

In [15], Ma and Nenghai Liu have proposed a method which calculates the usefulness of a packet in a node's buffer in terms of how well it can be delivered to the destination node. To achieve this, each packet in the node is assigned a weight W_i. The message weight is composed of two parts: one part takes into account the possibility that the packet reaches the destination directly and second part considers the possibility that it may be relayed to other nodes and then reach the destination. The first part Wd_i is calculated as

$$Wd_i = 1 - e^{-\lambda T}$$

where λ is the parameter associated with the inter contact time between the nodes which is exponentially distributed and T is the remaining TTL of the message. The second part of the weight is calculated using the information about how many copies of this packet have been distributed to the network. For this each node records the number of copies of a packet i it has created as NC_i and the total copy number of packet i as TC_i. During an encounter opportunity, each node updates its TC_i using the encountered node's NC_i.

$$Wi_i = \frac{NC_i - TC_i}{NC_i}$$

The total weight for a packet i is then calculated as: $W_i = \alpha Wd_i + \beta W_i$.

Whenever the buffer is full, the packet with the lowest W_i is dropped. This approach can work effectively in both sparse and dense networks. In dense networks, the nodes tend to relay more than directly deliver the packets and therefore the parameter Wi_i is important and in sparse networks where nodes can directly deliver the packets Wd_i are more important.

2.10 Congestion avoidance in a data centric opportunistic network

In [16], Bjurefors *et al.* have proposed dropping policy based on the interests of nodes in topics and the degree of replication of packets. The scheme was evaluated on Haggle project which itself is a data centric network architecture based on a publish/subscribe model. The strategies evaluated are least interested (LI), most interested (MI), max–max copies, most forwarded (MF), least forwarded (LF), random, infinite buffer and no buffer. In LI, when the buffer is full the packet in which least number of neighbours are interested in is deleted. Such a method increases the overall delivery rate in a data centric network because the overall interest of the network keeps increasing. This is sensible because the data in which no nodes are interested will eventually expire and may be dropped. MI performs just the opposite of LI by dropping the packets that most neighbours are interested in. This is judicious in the sense that it can reduce the number of copies of the packets which most nodes keep looking for. In max–max approach, the data objects are removed after they are copied a max number of times. This reduces the data objects being copied for a lot number of times. MF strategy drops the packet that has made the highest number of replications. The packet which has replicated many times must have reached many nodes and has a higher chance of reaching the destination and therefore is sensible to drop it first. LF operates just the opposite of MF by dropping the packet which has been least replicated. Infinite buffer policy assumes that each node has an infinite relay buffer and therefore no dropping policy is required, whereas no buffer policy assumes that nodes do not have a relay node and the data are accepted only if node is interested in it. The simulation results show that MF strategy is the best in terms of delivery ratio, delay and overhead.

2.11 Buffer management policy for DTN based on message size

In [17–19], message size is used as the criteria for buffer management. Drop largest [17] drops the message with the largest size. Thus, a large packet is sacrificed for preventing the drop of many small packets. In the T-drop [18], the packet which lies in the threshold range of the buffer is dropped. Equal drop (E-drop) [19] is a policy where a packet of the same size is dropped when the node is congested. This does not take care of the network condition to drop the packet. When a message has to be put into the buffer, E-drop chooses that another message of equal length be dropped to accommodate the new message. The advantage is that in some cases when a packet with a smaller size is deleted to make more space some other packets will have to be dropped additionally. Thus, for accommodating one packet we have to drop more packets. In E-drop such a situation is saved. The mean-drop policy [20] calculates the mean of the size of the messages in the congested node and then drops only those messages whose size are greater than or equal to the mean. These policies gave a new look into the buffer management policy by introducing a new parameter apart from just considering how old or how much replicas the message to be dropped has.

2.12 Buffer management for preferential delivery in opportunistic networks

The approach proposed in [21] considers the message priorities for the scheduling and dropping of messages. The messages are assigned three priorities: bulk, normal and expedited. The bulk messages have the lowest priority and the expedited have the highest with normal in a medium priority. The scheduling policy gives priority to the expedited messages whenever an encounter opportunity arises. When a packet arrives in the node, the bundle classifier stores them in their appropriate queue in terms of the class to which the message belongs to. When an encounter opportunity occurs the bundle scheduler is invoked to schedule the packets for forwarding. The bundle dropper drops the message by giving more priority to bulk and normal messages. The policy takes care that a node never drops a message that it has created. The scheduler transmits the bundles from high priority to low priority in a round robin fashion.

2.13 Buffer management scheme based on message transmission status

Liu *et al.* proposed a buffer management scheme inspired by the law of diminishing marginal utility in [22]. The law states that as a user increases consumption of a product, there is a decline in the marginal utility that user derives from consuming each additional unit of that product. More RCs of a message in the network is therefore not good as they consume the buffer space of other nodes preventing other messages to have a place in them.

The buffer management scheme consists of two parts: a buffer replacement scheme and a buffer scheduling scheme. The buffer management scheme works on two main properties of the message: the number of RCs of a message in a network and the dissemination speed of the message. The replication number of message i known to node m is denoted as R_m^i. Whenever node i meets node j that does not have message m, the replication number of m is calculated as:

1. Let node j be selected as a relay node by the routing protocol. Then if node j has no information about m, then replication

number of m is set as $R_m^i + 1$ in both the nodes. Otherwise it is set as $\max(R_m^i, R_m^j) + 1$ in both the sending and receiving nodes.
2. If node j is not selected as relay node, then if node j has no information about m the replication number is set as R_m^i on both sides. Otherwise it is kept as $\max(R_m^i, R_m^j)$.

The messages with lowest replication numbers are given high priority. When a message is to be dropped then the messages with lowest priority, that is, the ones having highest replication numbers are dropped. If all the messages have the same replication number then the dissemination speed of messages is used to break the tie. The dissemination rate of a message is calculated as

$$\text{rate} = \frac{K_m}{\text{TTL}_{\text{init}} - \text{TTL}}$$

where K_m is the number of hop counts experienced by message m. The messages with higher dissemination speeds are dropped first on a buffer overflow.

For scheduling purposes, all the messages selected by the routing protocol to be forwarded are sorted according to their priorities. Let this set of messages be called the 'ready set' (RS). If the lowest priority of messages in RS is greater than the highest priority of messages in the peering node, then all the messages are forwarded. If the highest priority of messages is lower than the lowest priority in the peering node, then only those messages are forwarded that can be accommodated in the free space. In all other cases, the RS and the peering node queue are merged.

This method is unique in the sense that it ensures fairness among the messages and thus it provides overall good performance in terms of delivery ratio and delay. This further underlines the fact that the number of RCs in the network is a good consideration for buffer management.

2.14 Enhanced buffer management policy that utilises message properties for DTN

Shin and Kim have proposed a buffer management policy that is also based on the message properties in [23]. This technique utilises three message properties namely the estimated number of replicas, the age and the remaining TTL of the message. For calculating the numbers of replicas of a message in the network, two parameters are used: the estimated total number of replicas (ETRs) and my forward (MF). Whenever node A meets node B and sends a message i to node B, then the MF_i value of node B is set to 0 and then B sets ETR_i value from node A

$$\text{ETR}_i^B \leftarrow \text{ETR}_i^A, \quad MF_i^B \leftarrow 0$$
$$\left(\text{ETR}_i^A \leftarrow \text{ETR}_i^A + 1, \quad MF_i^A \leftarrow MF_i^A \leftarrow MF_i^A + 1\right)$$

Now when node B meets node D which has the same message i, then the MF_i values of nodes B and D are exchanged so that both nodes can update the ETR_i value

$$\text{ETR}_i^B \leftarrow \text{ETR}_i^B + MF_i^D, \quad \text{ETR}_i^D \leftarrow \text{ETR}_i^D + MF_i^B$$

The ETR value of a message depicts how well the message is distributed in the network. Therefore a higher ETR value says that the message has more replicas in the network. The remaining TTL and age of a message show how much time the message has been in the network. The more the time spent in the network, the more is the chance to obtain its replicas distributed. In [23], two utility functions are defined.

$$\text{EBMP}_i^{\text{delivery}} = \frac{1}{\text{ETR}_i} + \frac{1}{\log(\text{AGE}_i)}\ \text{EBMP}_i^{\text{delay}}$$
$$= \frac{1}{\text{ETR}_i} + \log(\text{RTTL}_i)$$

The first utility function gives more emphasis on dropping the messages that has been replicated many number of times and the second function tends to delete an old message which has a shorter remaining TTL.

This strategy uses a better way to calculate the number of messages replicas by exchanging MF. In addition, it not only considers a single parameter, but additionally considers the age and remaining TTL to drop a message. However if two nodes meet frequently before meeting any other new nodes, there will be false updation of ETR values.

2.15 Buffer management policies in opportunistic networks

A novel buffer scheduling and dropping policy is proposed in [24] which uses the concept of average contact frequency among nodes for taking buffer management decisions. The message with the highest average contact frequency (ACF) value is duplicated to the peer and the message with the most replicas is dropped when the buffer is full. TheACF f_{ij} *is* defined as the number of encounters between node i and node j during a unit time period. ACF can be approximated with

$$f_{ij} = N_{ij}/T$$

where T is the pre-defined fixed length of time and N_{ij} is the number of contacts between i and j during T time.

When two nodes meet each other, they exchange those messages that are not in common in the following principle. The messages destined to the encountered node are firstly replicated to the peer in descending order of messages' creation time. Other messages that meet forwarding condition are replicated to the encountered node in the descending sequence of the ACF value. A dropping policy of messages based on the copies of messages is used. The copies of each message is calculated using a simple hop-count method.

3 Summary

From the above study of various buffer management policies proposed for DTNs, it can be concluded that various factors which affect the buffer dropping and scheduling decision are:

- Remaining TTL of message (the amount of time left before the message dies).
- Hop count of message (number of hops a message has travelled from the source node to the current node).
- Number of replicas of a message (number of nodes in the network having a message copy).
- Size of the message.
- Age of a message (time since the message has been created).
- Delivery cost (e.g. delivery predictability of message).
- Service count (number of transmission events for a message).
- Distance to the destination (distance between the current buffer node and destination node of a buffer).

Out of these factors hop count, number of replicas of a message and forward count properties of a message signify how well a message has been distributed in the network. A message which has high value for these parameters means it has been spread well. Remaining TTL of message and age signify the amount of

time a message has spent in the network. A message with smaller RTTL or large age means it has been in the network for a long time and should have been distributed well. Size of the message signifies the amount of buffer space occupied by the message or the amount of bandwidth the message would occupy during a transmission opportunity. Delivery cost or distance to destination parameters signify how far a message is from its destination.

Basically buffer management policies can be divided into three types:

- Global buffer management policy which utilise network-wide information regarding all messages.
- Local buffer management policy which use partial network knowledge like number of copies of message in the network, instead of all network-wide information correlated with messages and additional message properties like remaining TTL, size etc.
- Traditional buffer management policies like drop head, drop tail drop random.

Traditional buffer management policies perform poorly in DTN environments. Although global buffer management schemes utilising network-wide information all outperform the ones based on local information in respects of optimising specific network performance; policies based on partial network information can compensate their limitations and they can improve the routing metrics to near optimal levels.

4 Conclusion

In this paper, a survey of various buffer management schemes proposed in the literature for delay tolerant networks has been conducted. As DTNs use store and forward method for forwarding the messages to the destination, buffer scheduling and dropping policies play a very important role in the efficient delivery of messages. It may be seen that a number of message parameters have been used in different works for efficient buffer management; like remaining TTL of the message, hop count of the message, number of replicas of a message, size of a message, age of a message, delivery cost, forward count, distance to the destination etc. Most of the works have given emphasis to the number of replicas a message has. However, determining the exact number of replicas of a message is not easy since the network is quite distributed. Each method has used a unique way to determine the number of replicas. It may also be observed that a combination of multiple message properties can result in better buffer management decisions. Buffer management schemes in a DTN should be designed considering the limited storage of nodes and the short contact duration between nodes. In this paper we have tried to present a variety of buffer management schemes that are generic for any routing protocol and specific for some routing protocols. It would be interesting to combine various message scheduling and message dropping policies and study their effects on various routing protocols designed for DTN. Use of soft computing techniques for taking buffer management decisions would be another interesting future direction to work.

5 References

[1] Cao Y., Sun Z.: 'Routing in delay/disruption tolerant networks: a taxonomy, survey and challenges', *IEEE Commun. Tutor.*, 2012

[2] Delay Tolerant Networking Research Group. Available at http://www.dtnrg.org

[3] Delay Tolerant Network: Challenges and Application by Farid Farahmand (AITISl)

[4] Krifa A., Barakat C., Spyropoulos T.: 'Optimal buffer management policies for delay tolerant networks'. IEEE Conf. Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2008), June 2008, pp. 1–9

[5] Davis J.A., Fagg A.H., Levine B.N.: 'Wearable computers as packet transport mechanisms in highly partitioned ad-hoc networks'. Int. Symp. Wearable Computing, 2001, pp. 141–148

[6] Burgess J., Gallagher B., Jensen D.: 'MaxProp: routing for vehicle-based disruption-tolerant networks'. INFOCOM 2006 – The 25th IEEE Int. Conf. Computer Communications, Barcelona, Catalunya, Spain, 23–29 April 2006, pp. 1–11

[7] Vahdat A., Becker D.: 'Epidemic routing for partially connected ad hoc networks'. Technical Report, CS-200006, Duke University, 2000

[8] Lindgren A., Doria A., Schelen O.: 'Probabilistic routing in intermittently connected networks', *SIGMOBILE Mob. Comput. Commun. Rev.*, 2003, **7**, (3), pp. 19–20

[9] Spyropoulos T., Psounis K., Raghavendra C.S.: 'Spray and wait: an efficient routing scheme for intermittently connected mobile networks'. ACM SIGCOMM 2005 – Workshop on Delay Tolerant Networking and Related Networks (WDTN-05), Philadelphia, PA, USA, 22–26 August 2005, pp. 252–259

[10] Lindgren A., Phanse K.S.: 'Evaluation of queuing policies and forwarding strategies for routing in intermittently connected networks'. Proc. IEEE COMSWARE, January 2006, pp. 1–10

[11] Ramanathan R., Hansen R., Basu P.: 'Prioritized epidemic routing for opportunistic networks'. ACM MobiOpp'07, PR, USA, June 2007, pp. 62–66

[12] Li Y., Qian M.: 'Adaptive optimal buffer management policies for realistic DTN'. IEEE Global Telecommunication Conf., GLOBECOM 2009, USA, 30 November–4 December, 2009, pp. 1–5.

[13] Li Y., Zhao L., Liu Z., Liu Q.: 'N-drop congestion control strategy under epidemic routing in DTN'. Fifth ACM Conf. Mobile Computing Conf. (IWCMC 2009), Leipzig, Germany, 21–24 June 2009, pp. 457–460

[14] Vasco N.G.J., Farahmand S.F., Rodrigues J.J.P.C.: 'Performance analysis of scheduling and dropping policies in vehicular delay-tolerant networks', *Int. J. Adv. Internet Technol.*, 2010, **3**, (1), pp. 137–145, ISSN: 1942---2652, http://www.iariajournals.org/internet_technology

[15] Ma K.Y., Nenghai Liu B.: 'A new packet dropping policy in delay tolerant network'. USTC, Information Process Center 4th mailbox Hefei city, Anhui province, China, pp. 377–380

[16] Bjurefors F., Gunningberg P., Rohner C., Tavakoli S.: 'Congestion avoidance in a data-centric opportunistic network'. ACM ICN'11, ON, Canada, 2011, pp. 32–37

[17] Rashid S., Ayub Q.: 'Effective buffer management policy DLA for DTN routing protocols under congestion', *Int. J. Comput. Netw. Secur.*, 2010, **2**, (9), pp. 118–121

[18] Ayub Q., Rashid S.: 'T-drop: an optimal buffer management policy to improve QOS in DTN routing protocols', *J. Comput.*, 2010, **2**, (10), pp. 46–50

[19] Rashid S., Ayub Q., Soperi Mohd Zahid M., Abdullah A.H.: 'E-Drop: an effective drop buffer management policy for DTN routing protocols', *Int. J. Comput. Appl.*, 2011, **13**, (7), pp. 8–13

[20] Rashid S., Abdullah A.H., Soperi Mohd Zahid M., Ayub Q.: 'Mean drop an effectual buffer management policy for delay tolerant network', *Eur. J. Sci. Res.*, 2012, **70**, (3), pp. 396–407

[21] Fathima G., Wahidabanu R.S.D.: 'Buffer management for preferential delivery in opportunistic delay tolerant networks', *Int. J. Wirel. Mob. Netw., AIRCC Publ.*, 2011, **3**, (5), pp. 15–28

[22] Liu Y., Wang J., Zhang S., Zhou H.: 'A buffer management scheme based on message transmission status in delay tolerant networks', *IEEE Globecom*, 2011, pp. 1–5

[23] Shin K., Kim S.: 'Enhanced buffer management policy that utilises message properties for delay-tolerant networks', *IET Commun.*, 2011, **5**, (6), pp. 753–759

[24] Tang L., Chai Y., Li Y., Weng B.: 'Buffer management policies in opportunistic networks', *J. Comput. Inf. Syst. Binary Inf. Press*, 2012, **8**, (12), pp. 5149–5159