A Buffer Management Strategy on Spray and Wait Routing Protocol in DTNs

En Wang¹², Yongjian Yang¹, Jie Wu² and Wenbin Liu³

1 Department of Computer Science and Technology, Jilin University, Changchun, China
2 Department of Computer and Information Sciences, Temple University, Philadelphia, USA
3 Department of Software Engineering, Jilin University, Changchun, China

wangen0310@126.com ; yyj@jlu.edu.cn ; jiewu@temple.edu ; nike_lwb@126.com
Outline

• 1. Introduction
• 2. Model Description
• 3. Scheduling and Drop Strategy
• 4. Evaluation
• 5. Future Work
1. Introduction

1.1 Motivation

- The **dramatic change** of topology and the **frequent interruption** of connections make it difficult to forward the message to the destination in DTNs

- To maximize delivery ratio, while reducing the network congestion, **Spray and Wait** adopts a **binary** splitting method to distribute a set number of copies into the network

- However, there is still **partial congestion** due to the **limited buffer size**, even in the Spray and Wait routing protocol.
1. Introduction

1.2 Problem

- How to address the message scheduling and drop problem in Spray and Wait routing protocol ($M$: message id, $C$: message copies number, $R$: message remaining TTL).
1. Introduction

1.2 Problem

- When a connection is established, which message to send first
- When overflow occurs, which message to drop
1. Introduction

1.3 Challenge

- In order to optimize the delivery ratio, how to decide the message priority.

- How to map the number of copies \( C_i \) and remaining TTLs \( R_i \) into message priority

- How to address message scheduling and drop problem according to the priority.
Outline

- 1. Introduction

- **2. Model Description**

- 3. Scheduling and Drop Strategy

- 4. Evaluation

- 5. Future Work
2. Model Description

2.1 Mobility Model

• **Definition 1**: Intermeeting time: the elapsed time from the end of the previous contact to the start of the next contact between nodes in a pair

• **Definition 2**: Minimum intermeeting time: the minimum elapsed time for a specific node from the end of the previous contact to the start of the next contact with any other node.

• Intermeeting times are exponentially distributed under many popular mobility patterns such as random walk, random waypoint, and random direction.

\[ f(x) = \lambda e^{-\lambda x} \quad (x \geq 0) \]
2. Model Description

2.1 Mobility Model: the random-waypoint (a) and the real trace EPFL (b)
## 2. Model Description

### 2.2 Utility Model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Total number of nodes in the network</td>
</tr>
<tr>
<td>$K_{(t)}$</td>
<td>Number of distinct messages in the network at time $t$</td>
</tr>
<tr>
<td>$TTL_i$</td>
<td>Initial time-to-live (TTL) for message $i$</td>
</tr>
<tr>
<td>$R_i$</td>
<td>Remaining time-to-live (TTL) for message $i$</td>
</tr>
<tr>
<td>$T_i$</td>
<td>Elapsed time for message $i$ since its generation $(T_i = TTL_i - R_i)$</td>
</tr>
<tr>
<td>$n_i(T_i)$</td>
<td>Number of nodes with message $i$ in buffer after elapsed time $T_i$</td>
</tr>
<tr>
<td>$m_i(T_i)$</td>
<td>Number of nodes (excluding source) that have seen message $i$ after elapsed time $T_i$</td>
</tr>
<tr>
<td>$d_i(T_i)$</td>
<td>Number of nodes that have dropped message $i$ after elapsed time $T_i$ $(d_i(T_i) = m_i(T_i) + 1 - n_i(T_i))$</td>
</tr>
<tr>
<td>$E(I)$</td>
<td>Mathematical expectation of intermeeting times</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Parameter in the exponential distribution of intermeeting times $(\lambda = \frac{1}{E(I)})$</td>
</tr>
<tr>
<td>$E(I_{min})$</td>
<td>Mathematical expectation of the minimum intermeeting times</td>
</tr>
<tr>
<td>$\lambda_{min}$</td>
<td>Parameter in the exponential distribution of minimum intermeeting times $(\lambda_{min} = \frac{1}{E(I_{min})})$</td>
</tr>
<tr>
<td>$C$</td>
<td>The initial number of copies of message $i$ in source node</td>
</tr>
<tr>
<td>$C_i$</td>
<td>The copies number of message $i$ in the current node</td>
</tr>
<tr>
<td>$U_i$</td>
<td>Priority of message $i$</td>
</tr>
<tr>
<td>$P(T_i)$</td>
<td>Probability that message $i$ has been successfully delivered after elapsed time $T_i$</td>
</tr>
<tr>
<td>$P(R_i)$</td>
<td>Probability that undelivered message $i$ will reach the destination within time $R_i$</td>
</tr>
<tr>
<td>$P_i$</td>
<td>Probability that message $i$ can be successfully delivered</td>
</tr>
<tr>
<td>$P$</td>
<td>Global delivery ratio</td>
</tr>
</tbody>
</table>
2. Model Description

2.2 Utility Model

- The probability of message $i$ being delivered is given by the probability that message $i$ has been delivered and the probability that message $i$ has not yet been delivered, but will be delivered during the remaining time $R_i$

$$P_i = (1 - P_{T_i})P_{R_i} + P_{T_i}$$
2. Model Description

2.2 Utility Model

• Due to the reason that all the nodes including the destination have an equal chance of seeing the message $i$:

\[ P(T_i) = \frac{m_i(T_i)}{N - 1} \]

• Probability that undelivered message $i$ will reach the destination within time $R_i$:

\[
P(R_i) = 1 - \prod_{k=0}^{\log_2 C_i} e^{-\lambda n_i(T_i)[R_i - kE(I_{\text{min}})]}
\]

\[
= 1 - e^{-\lambda n_i(T_i)[(\log_2 C_i + 1)R_i - \frac{1}{2(N-1)\lambda} \log_2 C_i (\log_2 C_i + 1)]}
\]
2. Model Description

2.2 Utility Model

• We obtain the final expression for $P_i$ as follows:

\[
P_i = \frac{m_i(T_i)}{N - 1} + \left(1 - \frac{m_i(T_i)}{N - 1}\right) \\
(1 - e^{-\lambda n_i(T_i)}[(\log^2_i + 1)R_i - \frac{1}{2(N-1)\lambda} \log^2_i (\log^2_i + 1)])
\]

• Note that the global delivery ratio $P$ equals the sum of $P_i$:

\[
P = \sum_{i=1}^{K(t)} \left[ \frac{m_i(T_i)}{N - 1} + \left(1 - \frac{m_i(T_i)}{N - 1}\right) \\
(1 - e^{-\lambda n_i(T_i)}[(\log^2_i + 1)R_i - \frac{1}{2(N-1)\lambda} \log^2_i (\log^2_i + 1)]) \right]
\]
2. Model Description

2.2 Utility Model

• Three cases:

\[
\begin{align*}
\Delta(n_i) &= 1 & \text{If replicate message } i \text{ during contact.} \\
\Delta(n_i) &= 0 & \text{If no action for message } i \text{ is taken.} \\
\Delta(n_i) &= -1 & \text{If drop an already existing message } i.
\end{align*}
\]

• Therefore, the utility of message \( i \) is precisely the derivative of the delivery ratio \( P_i \), which is defined as \( U_i \):

\[
U_i = (1 - \frac{m_i(T_i)}{N - 1}) \lambda [(\log_2^{C_i} + 1) R_i - \frac{1}{2(N - 1)\lambda} \log_2^{C_i} (\log_2^{C_i} + 1)]
\]

\[
e^{-\lambda n_i(T_i) [(\log_2^{C_i} + 1) R_i - \frac{1}{2(N - 1)\lambda} \log_2^{C_i} (\log_2^{C_i} + 1)]}
\]

• The higher \( U_i \) indicates that the message \( i \) is more important
2. Model Description

2.2 Utility Model

• Estimation of \( m_i(T_i) \) and \( n_i(T_i) \):

\[
n_i(T_i) = m_i(T_i) + 1 - d_i(T_i)
\]

• \( d_i(T_i) \) is achieved as follows:

![Data Structure Table]

<table>
<thead>
<tr>
<th>Node A</th>
<th>Node B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Dropped List</th>
<th>Record Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Messages:1,3,5</td>
<td>300</td>
</tr>
<tr>
<td>B</td>
<td>Messages:1,4,6</td>
<td>350</td>
</tr>
<tr>
<td>C</td>
<td>Messages:2,7,8</td>
<td>250</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Dropped List</th>
<th>Record Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Messages:1,3,5</td>
<td>300</td>
</tr>
<tr>
<td>B</td>
<td>Messages:1,4,6</td>
<td>350</td>
</tr>
<tr>
<td>C</td>
<td>Messages:2,7,8</td>
<td>250</td>
</tr>
</tbody>
</table>
2.2 Utility Model

- \( m_i(T_i) \) is estimated as follows:

\[
m_i(T_i) = \sum_{k=1}^{n-1} 2^{\left\lfloor \frac{t_n - t_k}{E(T_{min})} \right\rfloor} + 1
\]
Outline

• 1. Introduction
• 2. Model Description
• 3. Scheduling and Drop Strategy
• 4. Evaluation
• 5. Future Work
3. Scheduling and Drop Strategy

3.1 Strategy

Algorithm 1 SDSRP

Input:
- Copies number: \( C \)
- Remaining \( TTL \): \( R \)
- Number of messages in the buffer: \( n \)
- The \( ID \) of new coming message: \( m \)

Output:
- Scheduling message: \( IDS \)
- Dropping message: \( ID_D \)

1. for \( i = 1 \) to \( n \) do
2. map \( C_i, R_i \) to \( Priority_i \)
3. Sort \( Priority_i \) incrementally
4. Find highest \( Priority_h \), and assign \( h \) to \( IDS \)
5. Find lowest \( Priority_l \), and assign \( l \) to \( ID_D \)
6. if connection up then
7. return \( IDS \)
8. if buffer overflows then
9. map \( C_m, R_m \) to \( Priority_m \)
10. if \( Priority_m < Priority_l \) then
11. assign \( m \) to \( ID_D \)
12. return \( ID_D \)
Outline

• 1. Introduction
• 2. Model Description
• 3. Scheduling and Drop Strategy
• 4. Evaluation
• 5. Future Work
4. Evaluation

4.1 Simulation parameters (random-waypoint)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Random-Waypoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Time</td>
<td>18000s</td>
</tr>
<tr>
<td>Simulation Area</td>
<td>4500m×3400m</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>100</td>
</tr>
<tr>
<td>Moving Speed</td>
<td>2m/s</td>
</tr>
<tr>
<td>Transmission Speed</td>
<td>250Kbps</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>100m</td>
</tr>
<tr>
<td>Buffer Size</td>
<td>2MB,2.5MB,3MB,3.5MB,4MB,4.5MB,5MB</td>
</tr>
<tr>
<td>Message Size</td>
<td>0.5MB</td>
</tr>
<tr>
<td>Message generation rate</td>
<td>[10,15][15,20][20,25]…[35,40][40,45][45,50]</td>
</tr>
<tr>
<td>TTL</td>
<td>300mins</td>
</tr>
<tr>
<td>Initial Copies Number</td>
<td>16,20,24,28,32,36,40,44,48,52,56,60,64</td>
</tr>
</tbody>
</table>
4. Evaluation

4.2 Four buffer management strategies

1. Spray and Wait adopts the FIFO (first in first out) buffer management strategy.

2. Spray and Wait-O regards the ratio between the remaining TTL and initial TTL as the priority.

3. Spray and Wait-C treats the ratio between the current message copies number and initial copies number as the priority.

4. SDSRP is our method, use $U_i$ as the priority.
4. Evaluation

4.3 Simulation Results
4. Evaluation

4.3 Simulation Results
4. Evaluation

4.3 Simulation Results
### 4.4 Simulation parameters (EPFL)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>EPFL-Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Time</td>
<td>18000s</td>
</tr>
<tr>
<td>Number of Nodes</td>
<td>200</td>
</tr>
<tr>
<td>Transmission Speed</td>
<td>250Kbps</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>100m</td>
</tr>
<tr>
<td>Buffer Size</td>
<td>2MB, 2.5MB, 3MB, 3.5MB, 4MB, 4.5MB, 5MB</td>
</tr>
<tr>
<td>Message Size</td>
<td>0.5MB</td>
</tr>
<tr>
<td>Message generation rate</td>
<td>[10, 15], [15, 20], [20, 25], ... , [35, 40], [40, 45], [45, 50]</td>
</tr>
<tr>
<td>TTL</td>
<td>300mins</td>
</tr>
<tr>
<td>Initial Copies Number</td>
<td>16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64</td>
</tr>
</tbody>
</table>
4. Evaluation

4.5 Simulation Results
4. Evaluation

4.5 Simulation Results
4. Evaluation

4.5 Simulation Results
Outline

• 1. Introduction

• 2. Model Description

• 3. Scheduling and Drop Strategy

• 4. Evaluation

• 5. Future Work
1. Future Work

- Other replication-based routing schemes
  - delegation forwarding, etc

- The problem of messages in different sizes
Thank You