
1Dr.Hanumanthappa J, 2Mr.Annaiah H
1Dept of Studies in CS, Manasagangotri, University of Mysore, Mysore-570006,
2Dept of CS & Engg, Govt Engg College, Hassan, Karnataka, India.

Abstract—BD-SIIT and DSTM are widely deployed IPv4/IPv6 Transition mechanism to improve the performance of the computer network in terms of Throughput, End to End Delay (EED) and Packet Drop Rate (PDR). In this journal paper we have implemented and compared the Performance Issues of our newly proposed Dollops Wise Curtail (DW&C) IPv4/IPv6 Migration Mechanism with BD-SIIT and DSTM in NS2. Implementation and Comparison Performance Analysis between Dollops Wise Curtail, BD-SIIT and DSTM shows that Dollops Wise Curtail IPv4/IPv6 migration algorithm performance outperforms than BD-SIIT and DSTM. Based on extensive simulations, we show that DW&C algorithm reduces the Packet Drop Rate (PDR), End to End Delay (EED) and achieves better Throughput than BD-SIIT and DSTM. In our research work observation, the performance metrics such as Throughput, EED and PLR for DW&C, BD-SIIT and DSTM are measured using TCP, UDP, FTP and CBR Traffics.

Keywords: BD-SIIT, DSTM, DW&C etc.

I. Introduction

Divide and Conquer is an important data structure technique which is mainly used to implement wide range of problems such as Transition issues/Routing issues, Mobility issues, Sensor Networking issues, MANET’s issues, Performance Analysis of Protocols issues etc.[1][2][4][5]. The Divide-and-Conquer paradigm is mainly used to solve efficient and effective algorithms in IPv4/IPv6 migration mechanism. The Divide and Conquer technique plays an important key role in all different types of sorting/searching mechanisms such as Heap Sort, Selection Sort, Bubble Sort, Radix Sort, Quick sort, Merge Sorting etc.

We can find a couple of works carried out in this direction. Hanumanthappa, J, Manjaiyah, D. H. (2006) have implemented Divide and Conquer based DST/List Ranking/Slice Wise Curtail based techniques and Binary Search, Linear Search, Ternary Search etc. Divide-and-Conquer algorithms are naturally designed and implemented as recursive procedures[24][25]. Divide and conquer algorithms can also be implemented by a non-recursive algorithm that stores the partial sub-problems in some explicit data structure, like a stack, queue, priority queue, double ended queue (Deque) to design the Dollops Wise Curtail (DW&C). The Dollops Wise Curtail (DW&C) implemented in this way is also called basic DW&C[30]. The Dollops Wise Curtail can also be implemented in software using NS2/OPNET++/OMNET++ Simulator etc[24][25].

II. Related Works

2.1. BD-SIIT

In this research work we have also proposed a new transition algorithm called BD-SIIT. The BD-SIIT(Stateless Internet Protocol/Internet control messaging Protocol Transition(SIIT)) is a novel IPv6 transition mechanism which allows IPv6 only hosts to speak to IPv4 only hosts. The BD-SIIT translator is one which is recognized as a stateless IP/ICMP translation, because the translator executes each translation individually without any reference to previously converted packets[1][24][25].

2.2. DSTM

The Dual stack transition mechanism(DSTM) is also mainly created to make IPv4 to IPv6 migration for an IPv6 dominant network which consists of both IPv4 and IPv6 configured links and can communicate with IPv4 and IPv6 nodes[12][13]. The DSTM is an IPv4 to IPv6 transition proposal based on the uses of IPv4 over IPv6 dynamic tunnels, and the temporary allocation of IPv4 global addresses to dual stack hosts. The DSTM is intended for IPv6 only networks in which hosts need to exchange information with other IPv4 hosts or applications[28].

IPv4/IPv6 Transition Mechanism. They also compared Performance Analysis of Divide and Conquer based DST/List Ranking techniques with our previously implemented BD-SIIT and DSTM. The simulation
results observed for SL&C, BD-SIIT and DSTM are clarified that SL&C works better than BD-SIIT and DSTM in terms of Throughput, EED and PDR[30].

Hira Sathu, Mohib A Shah(2009) have heuristically tested the three well known IP Transition techniques such as IPv6to4, IPv6in4, and Dual Stack. In their practical experiment two tunneling and a Dual Stack mechanisms were configured and repercussion of these mechanisms on Video Packets was observed. Comparison between video protocols illustrates that MPEG-2 was highly repercussion by tunneling mechanisms having almost the same amount of bandwidth wasted while MP4 was least repercussion by tunneling mechanism.

Eun-Young Park, Jae-Hwoon Lee et al(2009) have proposed and implemented the IPv4/IPv6 Dual Stack Transition Mechanism(4 to 6 DSTM). Which can work even in the case that hosts in the IPv4 network initiate connections with hosts in the IPv6 network[2][8].

Chiranjit Dutta and Ranjeet Singh(2012) have tested and heuristically reckoned two transition mechanisms namely IPv6/ IPv4 Tunneling and Dual-stack mechanism, as they relate to the performance of IPv6. They explore the repercussion of these approaches on end-to-end user application performance using metrics such as Throughput, Latency and Host CPU utilization. All experiments were deportmented using Three dual stack(IPv4/IPv6) routers, an IPv6 router and two ends–stations running Windows 7, loaded with a IPv4/IPv6 dual stack.

The structure of this research paper is as follows. In section 2 the proposed methodology is explained with a neat block diagram along with diminution tree. The Performance analysis methods and metrics are described in section 3. The simulation results are discussed in section 4, the section 5 illustrates discussion and observations and finally the paper is concluded in section 6.

III. Proposed Methodology

In this paper we have proposed and implemented Dollops Wise Curtail(DW&C) based IPv4/IPv6 Transition Mechanism using Divide and Conquer strategy[30]. One more Divide and Conquer methodology entitled Dollops-Wise Curtail(DW&C) is successfully installed to detect the IPv4/IPv6 transition performance issues in a proper manner. The block diagram of the newly proposed Divide and Conquer based DW&C Transition of IPv4 to IPv6 algorithm is shown in Fig.1. Which clearly depicts the different junctures. The steps are: (a) Divide and Conquer(D&C) based on Dollops Wise Curtail(DW&C) BD-SIIT technique[30].

Dollops Wise Curtail(DW&C) BD-SIIT Translation process

IPv6 source address Split 128 bits of IPv6 address into 8 dollops(hunks) of 16 bits. 1.IPv6 packet transmission
2.IPv6 to IPv4 mapping calculation and address mapping.
3.IPv6 in IPv4 Header translation.
4.IPv4 Packet Transmission.

IPv6 destination address

“The Fig.1”: Block Diagram of the Proposed Methodology(DW&C).

The Dollops Wise Curtail(DW&C) is started by dividing the 128 bits of IPv6 address into an 8 Dollops(pieces) called hunks[30]. The size of each hunk is 16 bits which is equivalent to one word size. In the research work, hunks are considered from the Left to Right[30]. The research work fixes the width of each hunk is 16 bits. However, with a clear observation it was mentioned that the size of the Dollops should be always 16 bits. Since the IPv6
address is divided into 8 Dollops from left to right direction[29][30]. The experimental research work also clarified that the mechanism of traversing an 128 bits source IPv6 address does not affect on the performance of the Dollops(Slab) Wise Curtail(DW&C) algorithm for IPv4/IPv6 transition mechanism[29][30].

```
<table>
<thead>
<tr>
<th>Bits 0</th>
<th>15 16</th>
<th>31 32</th>
<th>47 48</th>
<th>63 64</th>
<th>79 80</th>
<th>95 96</th>
<th>111 112</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slab1</td>
<td>Slab2</td>
<td>Slab3</td>
<td>Slab4</td>
<td>Slab5</td>
<td>Slab6</td>
<td>Slab7</td>
<td>Slab8</td>
<td></td>
</tr>
</tbody>
</table>
```

“Fig.2”: Slicing(Slabing) of the 128 bits IPv6 address into 8 stripes (hunks)

A. Working Methodology of Dollops Wise Curtail (DW&C)

The proposed algorithm is called to be Dollops Wise Curtail addresses and it is illustrated in Fig.3. The basic structure of this algorithm is called as succession (position) based tree which is designed by using Piece(Slabs) Classification[29][30]. The destination IPv6 address of the incoming packet is divided into 8 separate 16-bit long hunks (stripes) accordingly[29][30].

```
struct dwc{
    unsigned char ml;
    unsigned int strt;
    unsigned int stp;
    }dwc[8];
ml=total_size_of_IPv6_address;
div=ml/16;
md=ml%16;
for(z=0;z<div;z++)
{
    dwc[z].strt=prx[z].stp=prx[z];
    dwc[z].ml=16;
    dwc[div].strt=prx[div]>>(md<<1);
    dwc[div].stp=(dwc[div].start)(((1<<(md+1))-1);
    dwc[div].ml=md;
    for(z=div+1;z<8;z++)
    {
dwc[z].strt=0;
    dwc[z].stop=0xFFFF;
    dwc[z].ml=0;
}
prx[z]; the ith 16-bit value of an IPv6 address space.
```

```
Pseudocode-1: The Pseudo code of Dollops division for 128 bits of IPv6 address.
```

“Fig.3”: Decomposition of an IPv6 128 bits address into 8 Dollops based on 16 bits word length components.

IV. Performance Analysis Methods And Metrics

Performance analysis is an important criterion for designing both IPv4 and IPv6 network. The metrics we considered in this innovative research work are latency, throughput and Packet Drop Rate (PDR). The Discrete event simulator NS2 particularly popular in the performance analysis of protocols.
community is mainly considered as the common scaffolding to simulate the various existing and the proposed IPv4/IPv6 transition scenarios in the research work. NS2 simulates the IPv4/IPv6 transition performance queries by using the various performance parameters such as Throughput, End-to-End Delay (RTT) (Latency), Packet Loss (Drop) Rate and it presents one of the best case scenarios for the typical use of IPv4/IPv6 transition.

The Implementation of IPv4/IPv6 BD-SIIT translator is simulated by using NS2 simulator.

A. Performance Metrics

In order to evaluate the performance issues of DW&C, BD-SIIT and DSTM IPv4/IPv6 Transition mechanism we deploy three different categories of key parameters such as Throughput, End-to-End Delay, Jitter and Packet Drop Rate [21][26].

Throughput: Throughput is a ratio of the number of bits received at the destination work stations to the total simulation time [21][26].

\[
\text{Throughput} = \frac{\sum \text{Packets received} \times \text{packets size} \times 8\% \text{total simulation time}}{\times 100\%} \quad (i)
\]

Average End-to-End Delay: This is a difference between when the Packet ends it time and when the Packet starts it time.

\[
\text{Delay} = \sum (\text{Packet End Time} - \text{Packet Start Time}) \quad (ii)
\]

Packet Loss Rate (PLR)/Packet Drop Rate (PDR): Packet loss occurs during the communication between two or more hosts across the network. When two hosts exchange packets between their operating systems and some of the packets get dropped during the transmission due to overload which is called Packet Loss Rate. Most commonly, the packet gets dropped before the destination can be reached [21][26].

\[
\text{Packet Loss Rate/dropped rate (PLR/PDR)} = \frac{K_s}{K_r} \quad (iii)
\]

Where \(K_s\) is the amount of packet sent at source and \(K_r\) is the amount of packet received at destination.

Jitter: Jitter is one which is defined as fluctuation of end to end delay from one packet to a next connection flow packet [21][26].

\[
\text{Jitter (J)} = O_{i+1} - O_i \quad (iv)
\]

Where \(O_{i+1}\) is Obstruction of \(i+1\)th packet and \(O_i\) is the Obstruction of \(i\)th communication packet.

V. Simulation Results

We compared the Performance of D&C based DW&C with two different transport layer protocols TCP and CBR. In the experimental scenario we have compared the Performance Issues of D&C based DW&C with BD-SIIT, DSTM IPv4/IPv6 Transition mechanisms [21][26]. The experiment it consists of varying packet size of each datagram i.e 32, 64, 128, 256, 512, 768, 1024 bytes and the Maximum Queue Length (IFQ) is set to 50 packets etc. Each simulation executes for 200 sec of simulation time. The number of runs by varying the traffic and the researcher has computed the Throughput, Average End to End delay and Packet Drop Rate etc [21][26].

"Table 1": Average End-to-End Delay for IPv6 network with DSTM.

<table>
<thead>
<tr>
<th>SL.No</th>
<th>Packet Size(Bytes)</th>
<th>FTP End-to-End delay(ms)</th>
<th>CBR End-to-End delay(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>256</td>
<td>90</td>
<td>70</td>
</tr>
<tr>
<td>02</td>
<td>512</td>
<td>100</td>
<td>85</td>
</tr>
<tr>
<td>03</td>
<td>1000</td>
<td>105</td>
<td>93</td>
</tr>
<tr>
<td>04</td>
<td>1256</td>
<td>130</td>
<td>122</td>
</tr>
</tbody>
</table>
“Fig.4”: Calculation of Average (Mean) End-to-End Delay (EED) using FTP and CBR using an area graph.

“Fig.5”: Computation of DSTM mean End-to-End Delay (EED) using FTP and CBR with line graph.

The Table-2 shows the Average EED of DW&C and BD-SIIT using FTP and CBR. The Fig.4 shows the comparison of EED of DW&C, BD-SIIT using FTP and CBR and the Fig.5 shows the comparison of EED of DSTM, DW&C and BD-SIIT using FTP and CBR[11][19][21][26].

Table 2: Mean EED of DW&C and BD-SIIT using FTP and CBR.

<table>
<thead>
<tr>
<th>S.No</th>
<th>Packet Size(Bytes)</th>
<th>EED for DW&amp;C using FTP (ms)</th>
<th>EED for DW&amp;C using CBR (ms)</th>
<th>EED for BD-SIIT using FTP (ms)</th>
<th>EED for BD-SIIT using CBR (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>256</td>
<td>80</td>
<td>60</td>
<td>85</td>
<td>63</td>
</tr>
<tr>
<td>02</td>
<td>512</td>
<td>90</td>
<td>65</td>
<td>94</td>
<td>68</td>
</tr>
<tr>
<td>03</td>
<td>1000</td>
<td>94</td>
<td>82</td>
<td>98</td>
<td>89</td>
</tr>
<tr>
<td>04</td>
<td>1256</td>
<td>125</td>
<td>94</td>
<td>127</td>
<td>96</td>
</tr>
</tbody>
</table>
“Fig.6”: Comparison of EED of DW&C, BD-SIIT using FTP and CBR.

“Fig.7”: EED Comparison of DSTM, DW&C, BD-SIIT using FTP and CBR.

The Table-3 shows the Throughput comparison for DW&C, BD-SIIT and DSTM.

“Table 3”: Comparison of DW&C, BD-SIIT and DSTM Throughput.

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Packet Size(Bytes)</th>
<th>Throughput of DW&amp;C(%)</th>
<th>Throughput of BD-SIIT(%)</th>
<th>Throughput of DSTM(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>32</td>
<td>98.8</td>
<td>97.0</td>
<td>94.1</td>
</tr>
<tr>
<td>02</td>
<td>64</td>
<td>96.1</td>
<td>95.6</td>
<td>92.6</td>
</tr>
<tr>
<td>03</td>
<td>128</td>
<td>94.7</td>
<td>94.0</td>
<td>90.9</td>
</tr>
<tr>
<td>04</td>
<td>256</td>
<td>88.4</td>
<td>84.7</td>
<td>87.1</td>
</tr>
<tr>
<td>05</td>
<td>512</td>
<td>86.8</td>
<td>82.3</td>
<td>85.6</td>
</tr>
<tr>
<td>06</td>
<td>768</td>
<td>82.3</td>
<td>77.8</td>
<td>80.2</td>
</tr>
<tr>
<td>07</td>
<td>1024</td>
<td>76.6</td>
<td>74.2</td>
<td>74.5</td>
</tr>
</tbody>
</table>
“Fig.8”: Comparison of Throughput of DWC, DSTM, BD-SIIT transition mechanism using line graph.

“Table-4”: Calculation of PLR of DW&C, BD-SIIT and DSTM when the packet size varies from 32 bytes to 1024 bytes using TCP traffic.

<table>
<thead>
<tr>
<th>Sl.No</th>
<th>Packet size(Bytes)</th>
<th>PLR of DW&amp;C(%)</th>
<th>PLR of BD-SIIT(%)</th>
<th>PLR of DSTM(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>32</td>
<td>0.10</td>
<td>0.23</td>
<td>0.34</td>
</tr>
<tr>
<td>02</td>
<td>64</td>
<td>1.15</td>
<td>1.28</td>
<td>1.39</td>
</tr>
<tr>
<td>03</td>
<td>128</td>
<td>1.34</td>
<td>1.42</td>
<td>1.65</td>
</tr>
<tr>
<td>04</td>
<td>256</td>
<td>1.65</td>
<td>1.98</td>
<td>2.79</td>
</tr>
<tr>
<td>05</td>
<td>512</td>
<td>2.36</td>
<td>2.94</td>
<td>3.41</td>
</tr>
<tr>
<td>06</td>
<td>768</td>
<td>2.68</td>
<td>3.46</td>
<td>3.98</td>
</tr>
<tr>
<td>07</td>
<td>1024</td>
<td>4.58</td>
<td>4.80</td>
<td>4.94</td>
</tr>
</tbody>
</table>

“Fig.9”: Comparison of Throughput of DW&C, DSTM, BD-SIIT transition mechanism using line graph.

VI. Discussion and Observations

This section, now presents and discusses the results of our research work based on Dollops Wise Curtail (DW&C) IPv4/IPv6 transition mechanism. The results observed for DW&C, BD-SIIT and DSTM are showed that DW&C works better than BD-SIIT and DSTM in terms of Throughput, EED and PLR. In the first observation, the performance metrics such as Throughput, Average EED and PLR for DW&C, BD-SIIT and DSTM are measured using TCP and UDP Traffic. In the second observation...
the performance metrics such as Throughput, Mean EED, and PLR for DW&C, BD-SIIT and DSTM are measured using FTP and UDP protocols[11][19][21][26].

VII. Conclusions

The purpose of research work, was to identify the performance parameters of DW&C, BD-SIIT and DSTM using TCP, UDP, FTP, and CBR protocols. By empirically simulating NS-2 on the test-bed, the following conclusions are obtained as follows:

i. The outcome for Throughput indicates 98.8(%) when the packet size was 32 bytes and it varies up to 76.6(%) when the packet size was 1024 bytes. The Throughput of BD-SIIT varies from 97.0(%) to 74.2(%) and DSTM Throughput varies from 94.1(%) to 74.5(%) when the packet size increases from 32 bytes to 1024 bytes[11][19].

ii. End-to-End delay(EED) experienced in all three transition mechanism show that DW&C delay varies from 80(ms) to 125(ms) for FTP Traffic and 60(ms) to 94(ms) for CBR, the BD-SIIT delay varies from 85(ms) to 127(ms) for FTP and 63(ms) to 93(ms) for BD-SIIT when the packet size varies from 256 to 1256 bytes. The research work was observed least delay in DW&C and maximum delay in BD-SIIT[21][26].

iii. The Packet Loss Rate(PLR) observed on all the three different transition mechanism proves that DW&C PLR varies from 0.10(%) to 4.58(%) BD-SIIT PLR varies from 0.23(%) to 4.80(%) and DSTM PLR varies from 0.34(%) to 4.94(%) for TCP and UDP traffic and it rectify that performance of DW&C is better than BD-SIIT and DSTM in terms of PLR. By using FTP and CBR Traffic the DW&C PLR varies from 1.67(%) to 5.12(%) for FTP, 1.85(%) to 5.64(%) for CBR, the BD-SIIT PLR varies from 1.72(%) to 5.22(%) for FTP and 1.98(%) to 5.75(%) for CBR, the DSTM PLR varies from 1.86(%) to 5.42(%) for FTP and 2.00(%) to 5.82(%) for CBR when the packet size varies from 256 to 1256 bytes[19][21][26].

References

Proceedings Papers:


[6]. D. Libes, A Debugger for Tel applications, 1993, in Tel/Tk workshop.


Theses:


[33]. Martin Suchara, Reliable Internet Routing, doctoral diss, Princeton University, September, 2011.


Journal Papers:


Books:


[39]. Andrew S. Tanenbaum, Computer Networks, Fourth edition, 2005