

BAB IV

IMPLEMENTASI DAN EVALUASI

Pada bab ini dibahas tentang tahap implementasi serta evaluasi dari desain dan analisis sistem yang dijabarkan pada bab sebelumnya. Tahap implementasi menerangkan modul-modul terpenting dari sistem analisis kinerja yang dibuat dengan PHP. Evaluasi dari sistem menjelaskan tentang menu tampilan dari sistem analisis kinerja level komponen dan protokol jaringan yang dibuat. Tetapi sebelumnya, perlu dijelaskan kondisi lingkungan baik software maupun hardware yang digunakan untuk melakukan evaluasi.

4.1 Spesifikasi Sistem

4.1.1 Spesifikasi Hardware

Hardware yang digunakan untuk melakukan implementasi dan evaluasi mempunyai konfigurasi sebagai berikut :

- a. Server dengan spesifikasi teknis sebagai berikut :
 - i. Processor : Pentium IV. 3,4 GHz
 - ii. Memory : RAM 256 MB
 - iii. Harddisk : 40 GB
 - iv. Ethernet : 10/100 MBps
- b. *Client* untuk melakukan perhitungan dan analisis dengan spesifikasi teknis sebagai berikut :
 - i. Processor : Pentium III 800 Ghz
 - ii. Memory : RAM 128 MB

- iii. Harddisk : 20 GB
- iv. Ethernet : 10/100 Mbps

4.1.2 Spesifikasi Software

Kebutuhan Software yang diperlukan adalah :

- a. Sistem Operasi Windows 2000 Profesional
- b. Software Ethereal, untuk *capture* data trafik jaringan.
- c. Database MySQL, untuk menyimpan data dalam bentuk tabel.
- d. PHP, untuk develop program.
- e. Microsoft Management Console
- f. Web Server IIS
- g. JpGraph untuk membuat grafik.
- h. MySQLFront, untuk front end MySQL.

4.2 Instalasi dan Pengaturan Sistem

Petunjuk instalasi dan pengaturan sistem adalah sebagai berikut :

1. Instalasi dari sistem ini dimulai dengan menginstal IIS sebagai Web Servernya. IIS secara default ada pada CD installer Windows 2000 ke atas.



Gambar 4.1. Add Component IIS

Instalasi dilakukan dari Menu Start, Control Panel, Add/Remove Program, Add/Remove Windows Component seperti ditunjukkan pada Gambar 4.1.

Secara default IIS akan membuat folder C:/Inetpub/wwroot/.

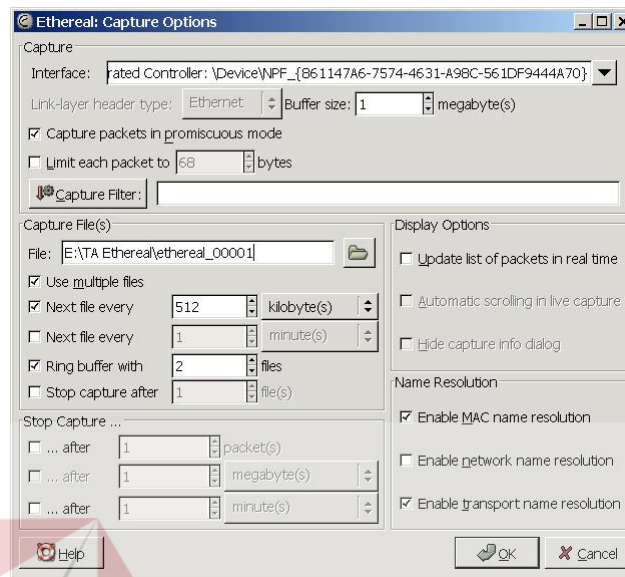
2. Setelah selesai melakukan instalasi web server, berikutnya adalah menginstal PHP sebagai *interpreter scripts*. PHP yang digunakan adalah PHP versi 4.3.10. Konfigurasi detail untuk PHP disediakan pada file php.ini.
3. Untuk membuat tampilah grafik, diinstall tool JpGraph. Yang pertama dilakukan adalah mengaktifkan library gd pada php.ini. Ekstrak file JpGraph dan pindahkan direktori src pada direktori web server (wwwroot).
4. Database yang digunakan adalah MySQL versi 3.23.36. Secara default MySQL akan membuat folder C:/mysql. Data tabel terletak pada folder C:/mysql/data/[nama database].
5. Instalasi Tool Packet Capture, yang diawali dengan install Winpcap.exe kemudian Ethereal.
6. Untuk memudahkan maintenance database MySQL bisa digunakan PHPMysqlAdmin atau MySQLfront.

4.3 Implementasi Prosedur

Prosedur yang harus dilakukan untuk implementasi yaitu :

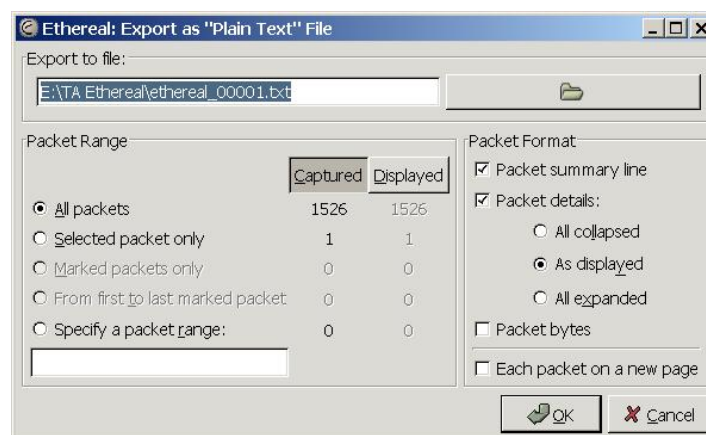
1. Setting log Ethereal seperti ditunjukkan pada Gambar 4.2. Setting Ethereal ini dilakukan pada server yang akan dihitung kinerjanya. Sistem Analisis Kinerja Level Komponen dan Protokol Jaringan yang dibuat hanya mampu menggunakan file log dengan ukuran maksimum 1 Mbyte. Hal ini disebabkan karena diperlukan waktu yang lama pada saat pemindahan

data dari file log ke tabel. Disarankan menggunakan file log dengan ukuran 512 Kbyte.



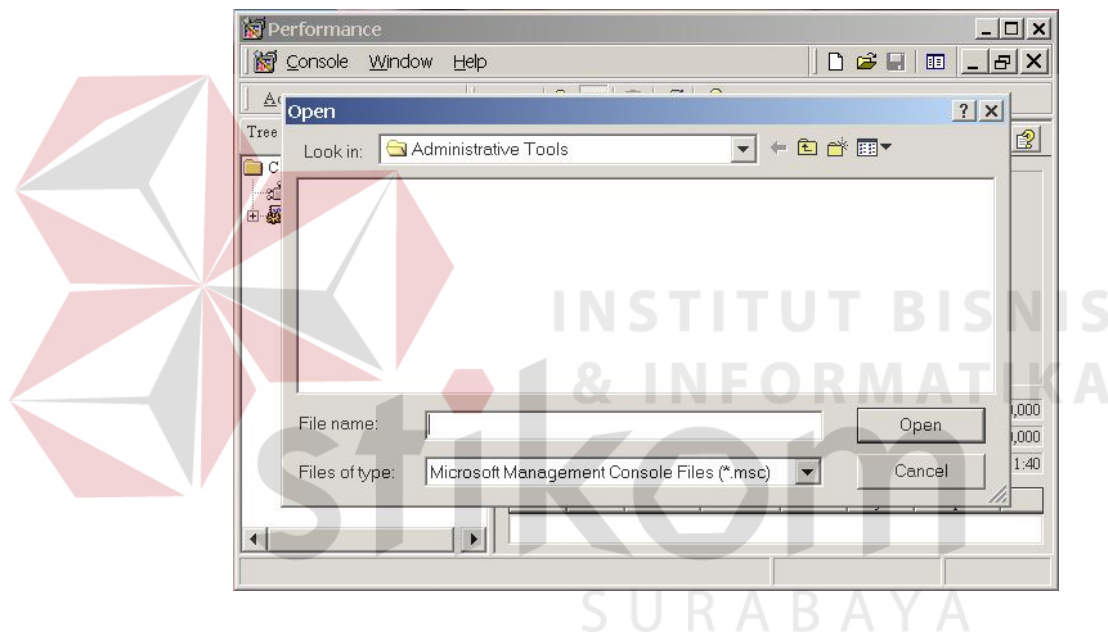
Gambar 4.2. Setting Ethereal untuk implementasi sistem

2. File log hasil capture dari Ethereal berformat tcpdump dan binary sehingga tidak bisa dibaca dengan teks editor. Untuk menghindari hal tersebut, Ethereal menyediakan menu untuk Export file hasil .log ethereal ke plain text, seperti tampak pada Gambar 4.3



Gambar 4.3. Ekspor File .log Ethereal ke Plain text

3. Setting file log pada MMC dengan membuka file .msc yang telah ditentukan (ada pada CD dalam buku ini), seperti tampak pada Gambar 4.4. Sistem Analisis Kinerja Level Komponen dan Protokol Jaringan Komputer ini hanya mampu memindahkan file log ke tabel dengan ukuran file maksimum 1 Mbyte. Disarankan menggunakan log file dengan ukuran 512 KByte.



Gambar 4.4. Open file .msc dari MMC

4. Data dari file log Ethereal yang dimasukkan ke sistem adalah file hasil ekspor ke plain text dengan ekstensi .txt. Sedangkan file log MMC yang digunakan berekstensi .csv, didapatkan langsung dari file log MMC.

4.4 Implementasi Sistem

Implementasi sistem menjelaskan tentang modul-modul penting dalam sistem yang dibuat. Beberapa modul tersebut adalah :

4.4.1 Upload File

Untuk menyimpan file ke web server dan atribut file ke tabel datafile digunakan function PHP. Modul ini menerima inputan berupa file, yang kemudian dipindahkan ke folder tertentu dalam web server, serta menyimpan nama file, ukuran file, tipe file ke tabel datafile. Gambar 4.5 menunjukkan modul untuk upload file tersebut. Detail lengkap modul ini ada pada lampiran dengan nama file input_log2.php.

```
<?php
$uploaddir = 'c:/inetpub/wwwroot/final/uploads/'; //folder tempat file disimpan
pada web server.

$uploadfile = $uploaddir.basename($_FILES['userfile']['name']); //nama file

//proses pemindahan file
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile)) {
    echo "File is valid, and was successfully uploaded.\n";
} else {
    echo "Possible file upload attack!\n";
}
$a = $_FILES['userfile']['name'];
$b = $_FILES['userfile']['size'];
$c = $_FILES['userfile']['type'];

if (preg_match ("/.txt/i", $a))
    $log="Ethereal";
else
    $log="MMC";

//memasukkan atribut file ke database
$dbQuery = "insert into datafile (id_datafile,nama, size, type, Jenis_log) values
($id_datafile,'$a','$b','$c','$log)";
mysql_query($dbQuery) or die("Couldn't add file to database Q1");
?>
```

Gambar 4.5 File untuk upload file ke database.

4.4.2 Interface Ethereal ke Database

Modul ini digunakan untuk memindahkan file hasil log Ethereal yang tidak beraturan ke database agar mudah untuk dilakukan perhitungan. Modul ini dimulai dengan membuka file, kemudian membaca tiap baris berurutan dari baris pertama sampai terakhir. Pada proses pembacaan baris tersebut dilakukan pula proses pencocokan, jika sesuai akan ditandai untuk dimasukkan ke tabel Frame. Pencocokan data dilakukan untuk mencari atribut no frame, waktu dilakukan capture, jenis protokol, ukuran packet, IP source, IP destination, MAC source, MAC destination, selang waktu antar packet. Sebagian isi modul ditunjukkan pada Gambar 4.6. Detail lengkap modul ini ada pada lampiran dengan nama file `input_frame.php`

```

$datafile="C:/inetpub/wwwroot/final/uploads/"._GET["a"];
$fp = fopen ($datafile,"r");
while ($data = fgets ($fp, 10000, "\t \t \t")) {
    $num = count($data);
    for ($c=0; $c < $num; $c++)
    {
        if (preg_match ("/Arrival Time/i", $data[$c]))
        {
            $waktu = explode(" ",$data[$c]);
            $tanggal = $waktu[2]." ".$waktu[3]." ".$waktu[4];

            //cari tanggal 1 digit dan 2 digit
            if (preg_match ("/,/i", $tanggal))
            {
                $tanggal=explode(",",$tanggal);
                $tanggal=$tanggal[0];
                print "Tanggal = ". $tanggal. "<br>\n"; // Tanggal
            }else{
                print "Tanggal = ". $tanggal. "<br>\n"; // Tanggal
            }
            $arr_time = $waktu[5]; //Arrival Time
        }
    }
}

```

Gambar 4.6 Sebagian modul untuk memindahkan file log Ethereal ke tabel

4.4.3 Interface MMC ke Database

Hampir sama dengan modul sebelumnya, modul ini digunakan untuk memindahkan file log MMC yang tidak beraturan untuk dipindahkan ke tabel-tabel yang telah disediakan. Isi modul ini adalah perintah untuk membaca file .csv kemudian melewati proses pemilihan field mana yang dipilih dan dimasukkan ke tabel. Sebagian isi dari modul ini tampak pada Gambar 4.7. Detail lengkap modul ini ada pada lampiran dengan nama file input_MMC.php

```

$datafile="C:/inetpub/wwwroot/final/uploads/" . $_GET["a"];
$fp = fopen ($datafile,"r");
while ($data = fgetcsv ($fp, 10000, "\t \t \t")) {
    $num = count($data);
    for ($c=0; $c < $num; $c++)
    {
        $pieces = explode ("|", $data[$c]);

        $waktu = $pieces[0];
        $committed_bytes_in_use = $pieces[1];
        $available_bytes = $pieces[2];
        $available_kbytes = $pieces[3];
        $available_mbytes = $pieces[4];
        $cache_bytes = $pieces[5];
        $cache_bytes_peak = $pieces[6];

        $dbQuery = "insert into memory values (0,$waktu,'
        $committed_bytes_in_use','$available_bytes','$available_kbytes','$available_mbytes',
        '$cache_bytes','$cache_bytes_peak','$cache_faults','$commit_limit','$committed_bytes',
        '$demand_zero_faults','$free_system_page_table_entries',
        '$page_faults','$page_reads','$page_writes','$pages_input','$pages_output','$pages',
        '$pool_nonpaged_allocs','$pool_nonpaged_bytes','$pool_paged_allocs','$pool_paged_bytes',
        '$pool_paged_resident_bytes','$system_cache_resident_bytes','$system_code_resident_bytes',
        '$system_code_total_bytes','$system_driver_resident_bytes','$system_driver_total_bytes',
        '$transition_faults','$write_copies)";

        mysql_query($dbQuery) or die("Couldn't add file to database Q1");
    }
}

```

Gambar 4.7 Sebagian modul untuk memindahkan file log MMC ke tabel

4.4.4 Perhitungan per Komponen

Modul ini berfungsi untuk melakukan perhitungan dari kinerja level komponen yang menggunakan metode Performance Modelling Concepts. Perhitungan dengan cara Little's Law, Forced Flow Law dan Utilization Law seperti ditunjukkan pada Gambar 4.8. Detail lengkap modul ini ada pada lampiran dengan nama file MMC.php

```
//Little's Law
$response_time_litte = $total_waiting_time/$jumlah_task_selesai;
$task_server = $total_waiting_time/$total_waktu;
$throughput_little = $jumlah_task_selesai/$total_waktu
...
//Forced Flow Law
$task_per_transaksi = $jumlah_task_selesai/$jumlah_transaksi_selesai;
$avg_transaksi = $jumlah_transaksi_selesai/$total_waktu;
$throughput_forced = $jumlah_task_selesai/$total_waktu;
...
//Utilization Law
$service_time = $waktu_sibuk/$jumlah_task_selesai;
throughput_utilization = $total_waktu/$jumlah_task_selesai;
$utilization = $jumlah_task_selesai/$total_waktu;
```

Gambar 4.8. Sebagian Modul untuk menghitung per komponen

4.4.5 Perhitungan per Protokol

Modul perhitungan per protokol digunakan untuk menghitung atribut protokol. Dimulai dengan menghitung jumlah secara keseluruhan ukuran frame yang lewat, kemudian dibedakan antara frame yang menuju server (received) dan dari server (send). Setelah itu, proses menandai jenis protokol dan jumlah ukuran frame pada masing-masing jenis protokol tersebut. Sebagian dari modul ini dapat

dilihat pada Gambar 4.9. Detail lengkap modul ini ada pada lampiran dengan nama file `ethereal.php`

```

$result = mysql_query("select * from frame") or die ("Invalid Query");

while ($row = mysql_fetch_row($result))
{
    $jumlah_row_frame = $jumlah_row_frame + 1;
    $bytes_packet_frame = $bytes_packet_frame + $row[5];
    $total_waktu_frame = $total_waktu_frame + $row[6];

    if($row[9]==$capex){
        $bytes_packets_frame_sent = $bytes_packets_frame_sent + $row[5];
        $jumlah_row_frame_sent = $jumlah_row_frame_sent + 1;
    }

    if($row[10]==$capex){
        $bytes_packets_frame_sent=$bytes_packets_frame_received + $row[5];
        $jumlah_row_frame_received = $jumlah_row_frame_received + 1;
    }

    if ($row[4]=='TCP')
    {
        $jumlah_row_tcp = $jumlah_row_tcp + 1;
        $bytes_packet_tcp = $bytes_packet_tcp + $row[5];
        $total_waktu_tcp = $total_waktu_tcp + $row[6];

        if($row[9]==$capex){
            $bytes_packets_tcp_sent = $bytes_packets_tcp_sent + $row[5];
            $jumlah_row_tcp_sent = $jumlah_row_tcp_sent + 1;
        }
    }
}

```

Gambar 4.9. Sebagian Modul untuk menghitung per protokol

4.4.6 Perhitungan per komponen per protokol

Modul terakhir yang akan dibahas adalah perhitungan per komponen per protokol. Modul ini mendapat input berupa modul perhitungan per komponen dan modul perhitungan per protokol. Hasil perhitungan per komponen akan dibagi secara rata-rata oleh komposisi jumlah protokol yang diperoleh dari modul

perhitungan per protokol. Potongan program tersebut, seperti tampak pada Gambar 4.10. Detail lengkap modul ini ada pada lampiran dengan nama file analisis.php.

```

$service_time = $total_bytes/$total_waktu;
<tr>
<td><b>Processor</b></td>
<td><?print
bcpow(($total_waktu_sibuk/$jumlah_task_selesai),1,6);?></td>
<td><?print
bcpow(((($total_waktu_sibuk/$total_waktu)*100),1,6);?></td>
<td><? print bcpow($residence_time_cpu,1,6);?></td>
</tr>
<tr>
<td>TCP Sent</td>
<td><?print bcpow( (($total_waktu_sibuk/$jumlah_task_selesai)
*($total_waktu_tcp_sent/$total_waktu_frame)/$jumlah_row_tcp_sent)
*1000000,1,6);?></td>
<td><?print bcpow((((($total_waktu_sibuk/$total_waktu)*100)*
($total_waktu_tcp_sent/$total_waktu_frame)/$jumlah_row_tcp_sent)*
1000000,1,6);?></td>
<td><? print bcpow(($residence_time_cpu*($total_waktu_tcp_sent/
$total_waktu_frame)/$jumlah_row_tcp_sent)*1000000,1,6);?></td>
</tr>

```

Gambar 4.10. Sebagian modul perhitungan per komponen per protokol

4.5 Evaluasi Sistem

Sistem ini dimulai dengan halaman login, untuk mempermudah identifikasi user yang melakukan akses terhadap sistem ini. Menu Login seperti tampak pada Gambar 4.11.

A login form with a blue border. It contains two input fields: one for 'User:' and one for 'Pass:'. Below the fields is a 'Submit' button.

Gambar 4.11. Menu Login

Setelah verifikasi user dan password, akan terdapat menu seperti ditunjukkan pada Gambar 4.12.

The main menu features a blue header with the title 'Analisis Kinerja Level Komponen & Protokol Jaringan'. Below the header is a navigation bar with the following items: Home, Input File log, Lihat Rekap File, Analisis MMC, Analisis Ethereal, Hasil Analisis, Grafik, and Logout. The main content area contains a welcome message for the Admin, a description of the system's purpose, and a list of menu items with their functions. A watermark for 'STIKOM SURABAYA' is visible in the background.

Selamat Datang Admin,
Sistem ini digunakan untuk melakukan analisis Kinerja Level Komponen dan Protokol Jaringan pada Server. Sistem ini mendapatkan input dari File hasil log Microsoft Management Console (MMC) dan Ethereal.

* Keterangan Menu

Input File .log	: Digunakan untuk menginputkan datafile dari hasil .log
Lihat Rekap File log	: Digunakan untuk melihat tabel di mana datafile log disimpan
Analisis MMC	: Menghitung Analisis dari MMC
Analisis Ethereal	: Menghitung Analisis dari Ethereal
Analisis	: Analisis Level Komponen per Protokol
Grafik	: Analisis Level Komponen per Protokol dalam Grafik

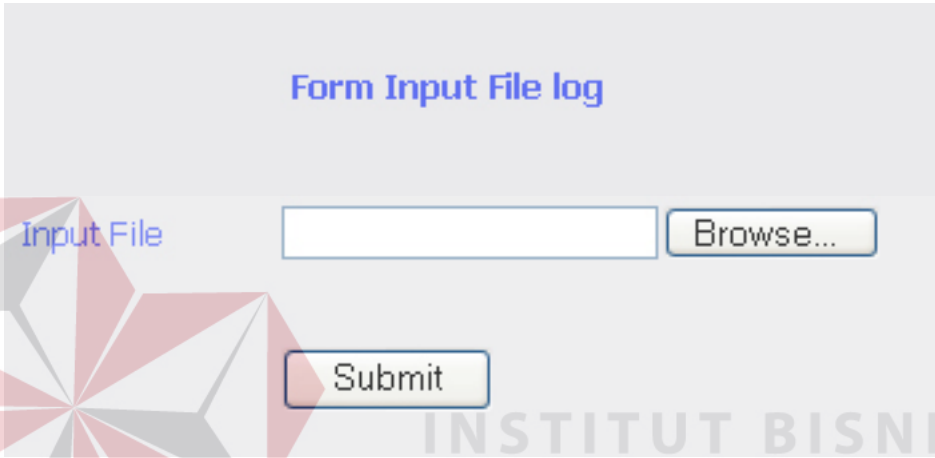
Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya

Gambar 4.12. Menu utama dan pilihan sub menu

Beberapa Menu yang tampak pada Gambar 4.12 adalah Input File log, Lihat Rekap File, Analisis MMC, Analisis Ethereal, Hasil Analisis, Grafik dan

Logout. Susunan Menu dibuat secara horizontal, karena tingkatan menu adalah sejajar.

Menu Input File log digunakan untuk menginputkan hasil file log dari MMC ataupun Ethereal ke dalam sistem untuk disimpan, seperti ditunjukkan pada Gambar 4.13.



Form Input File log

Input File

Browse...

Submit

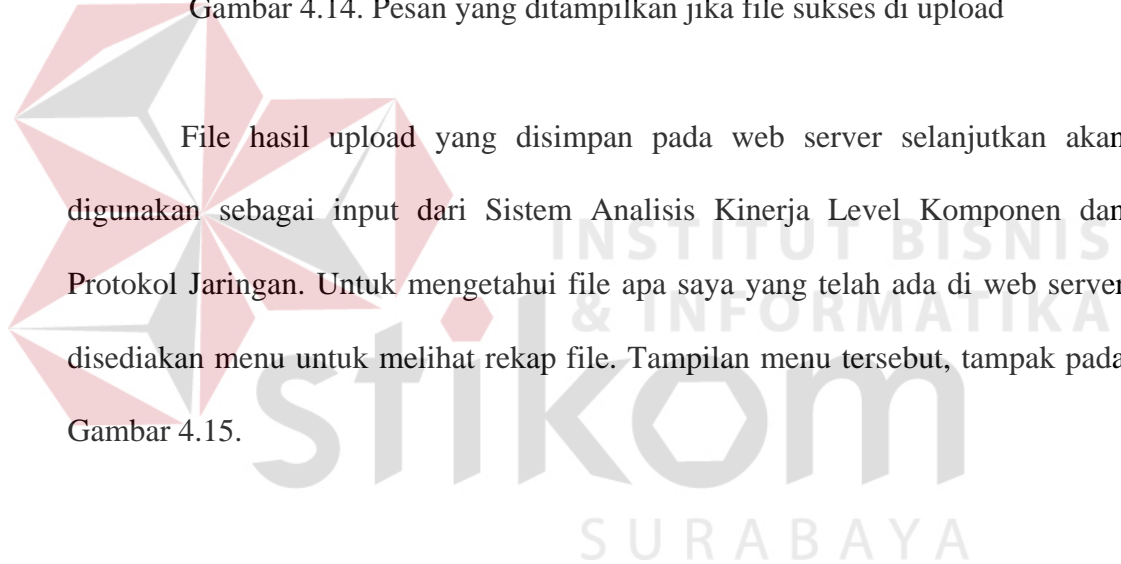
Gambar 4.13. Menu Input File log

Menu Input File log tersebut akan menyimpan file ke dalam web server pada folder yang telah ditentukan. Jika file sukses di upload, maka pesan yang akan ditampilkan ditunjukkan pada Gambar 4.14. Dari Gambar, dapat diketahui bahwa file telah sukses diupload. Informasi lain dari halaman ini, nama file dalam web server adalah mmc_000045.csv, sebelum dipindahkan ke web server untuk sementara file akan disimpan dulu di C:/PHP/uploadtemp/. Ukuran file MMC yang telah diupload adalah 524492 bytes atau 513,8 Kbyte. Untuk kemudahan dalam melakukan upload, disarankan untuk menggunakan file yang ukurannya kurang dari 1 Mbyte.

```
File is valid, and was successfully uploaded.
Here is some more debugging info:Array
(
    [userfile] => Array
        (
            [name] => mmc_000045.csv
            [type] => application/octet-stream
            [tmp_name] => C:\PHP\uploadtemp\php2.tmp
            [error] => 0
            [size] => 524492
        )
    )
)
```

Gambar 4.14. Pesan yang ditampilkan jika file sukses di upload

File hasil upload yang disimpan pada web server selanjutnya akan digunakan sebagai input dari Sistem Analisis Kinerja Level Komponen dan Protokol Jaringan. Untuk mengetahui file apa saja yang telah ada di web server disediakan menu untuk melihat rekap file. Tampilan menu tersebut, tampak pada Gambar 4.15.

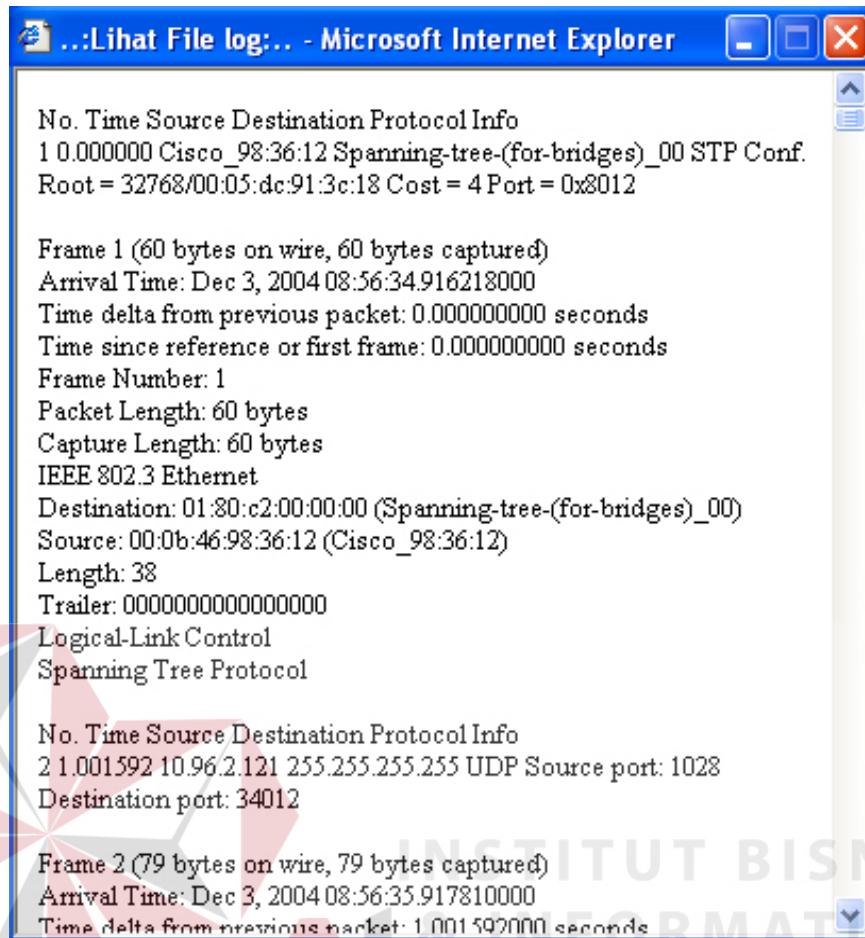


Rekap File log

No	Nama File	Dari File log	Frame	Server	Lihat	Hapus
1	test	MMC	✓	Input		
2	ethereal_00001.txt	Ethereal	Input	✓		
3	Ethereal_00002.txt	Ethereal	Input	✓		
4	Ethereal_00003.txt	Ethereal	Input	✓		
5	mnc_000001.csv	MMC	✓	Input		
6	mnc_000002.csv	MMC	✓	Input		
7	mnc_000003.csv	MMC	✓	Input		
8	mnc_000004.csv	MMC	✓	Input		
9	mnc_000012.csv	MMC	✓	Input		
12	mnc_000045.csv	MMC	✓	Input		

Gambar 4.15. Menu Rekap File .log

Untuk memasukkan file log ke dalam tabel dapat dilakukan pada menu rekap file, dengan cara klik pada Input Frame untuk file log Ethereal dan klik input Server untuk file log MMC. Jika dalam kolom Frame dan Server tertulis OK, maka menu tersebut tidak dapat dipilih, artinya dalam satu baris hanya terdapat satu menu Input dan satu menu OK. Pada menu rekap file log ini dapat juga digunakan untuk melihat isi file, seperti ditunjukkan pada Gambar 4.16.



Gambar 4.16. Isi file Ethereal

Isi dari file ethereal akan diolah oleh interface 1 untuk selanjutnya dimasukkan ke dalam Tabel Frame. Tabel Frame tersebut digunakan untuk menghitung kinerja per protokol jaringan. File MMC dipindahkan ke beberapa tabel, yaitu : disk, memory, process, processor dan sistem. File MMC dipindahkan oleh Interface 2. Tabel-tabel tersebut akan digunakan untuk menghitung kinerja pada level komponen.

Protokol	Total Frame	Total Frame Sent	Total Frame Received	Total Bytes Sent	Total Bytes Received	Total Bytes	Avg size/frame	Avg time/frame	Avg time/bytes	Avg bytes/time
All	83824	53504	17355	36891779	1732172	39607555	472.5085	0.3799	0.0008	1243.7497
TCP	17565	1952	15600	544980	1049569	1595343	90.8251	0.0165	0.0001	5501.0967

Protokol	Total Frame	Total Frame Sent	Total Frame Received	Total Bytes Sent	Total Bytes Received	Total Bytes	Avg size/frame	Avg time/frame	Avg time/bytes	Avg bytes/time
UDP	13521	13045	7	1060597	580	1139001	84.2394	0.769	0.0091	109.5426
ARP	367	199	110	11814	5646	20904	56.9591	0.2133	0.0037	267.0154
EIGRP	6883	0	0	0	0	509342	74	0.6742	0.0091	109.7495
STP	15924	11311	219	678660	13140	955440	60	0.888	0.0148	67.5644
HTTP	24956	23620	1331	34114740	653148	34772559	1393.3546	0.0102	0	135497.82
Others	4450	3269	45	376050	4458	503821	113.2182	0.4541	0.004	249.2877

Gambar 4.17. Contoh analisis Ethereal

Hasil Analisis dari file log Ethereal berupa data rekapitulasi protokol jaringan dalam melakukan proses terhadap frame yang datang. Contoh hasil rekap ditunjukkan pada Gambar 4.17, jumlah data yang digunakan sebanyak 83824 frame/record. Dari hasil analisis tersebut, dapat diketahui bahwa frame yang mempunyai ukuran rata-rata terbesar adalah protokol HTTP dengan 1393 bytes atau 1,36 Kbyte. Frame yang mempunyai ukuran rata-rata terkecil adalah frame dengan protokol ARP yaitu sebesar 56,9 bytes, urutan berikutnya STP dan EIGRP yang masing-masing sebesar 60 bytes dan 74 bytes.

Jumlah frame terbesar adalah HTTP sebanyak 24956 atau sebesar hampir 30% dari seluruh frame yang ter-capture. Jumlah frame terkecil adalah ARP sebanyak 367 atau sebesar 0,4 %.

Jumlah keseluruhan ukuran bytes pada protokol HTTP mencapai 34772 Kbyte atau sebesar 87% dari keseluruhan jumlah byte frame, sedangkan jumlah terkecil adalah ARP dengan 20 Kbyte atau sebesar 0,05%.

Waktu yang diperlukan untuk mengolah frame paling lama adalah pada protokol STP sebesar 0,88 sec setiap framenya. Waktu tercepat untuk mengolah frame pada protokol HTTP sebesar 0,01 sec setiap framenya. Dari segi waktu dan

jumlah frame dapat disimpulkan bahwa ukuran frame HTTP sangat besar tetapi diperlukan waktu tercepat untuk mengolahnya.

Menu Analisis MMC berupa data hasil perhitungan dari tabel disk, processor, memory, sistem dan process. Perhitungan ini didasarkan pada metode Performance Modelling Concepts.

PROCESSOR	
Jumlah Task (A)	6909
Total waktu (T)	35043.171
Processor Time (B)	1019.24942
Jumlah task selesai (C)	6879
Utilization of Server (U = B/T)	2.908 %
<i>Utilization Law</i>	
Rata-rata Service Time (S=B/C)	0.14816
Troughput (X = C/T)	0.19630
Utilization of Server (U = B/T)	2.908 %
<i>Little's Law</i>	
Rata-rata response time (R = W/C)	0.14816
Rata-rata Jumlah task pada server (n=W/T)	6.74908
Troughput Server (X=C/T)	0.19630
<i>Forced Flow Law</i>	
Visit Ratio Server (V(i) = C(i)/C(0))	19.43220
System Troughput (X(0)=C(0)/T)	0.01010
Server Troughput (X(i) = C(i)/T)	0.19630

Gambar 4.18. Hasil Perhitungan MMC untuk processor

Hasil perhitungan processor ditunjukkan pada Gambar 4.18. Dari Gambar ditunjukkan bahwa analisis processor dibedakan sesuai dengan hukum yang digunakan dalam perhitungan ini, yaitu : Little's Law, Forced Flow Law dan Utilization Law. Tidak semua perhitungan disini digunakan pada analisis residence time, hanya informasi dari hasil capture data MMC.

Perhitungan yang dilakukan pada menu analisis MMC berikutnya adalah disk dan incoming/outgoing link. Gambar 4.19 menunjukkan hasil perhitungan

pada disk dan incoming/outgoing link. Hasil perhitungan ini, tidak dibedakan berdasarkan metode Performance Modelling Concepts.

PHYSICAL DISK	
Waktu Sibuk (B)	120672 sec
Waktu Idle	3368548 sec
Rata rata file di retrieve /sec	4206.5124478958 byte
Request/sec	0.00733740649803
Kapasitas Disk C	42,35 %
Kapasitas Disk D	57,64 %
INCOMING/OUTGOING LINK	
Avg Bytes yang di receive/sec	73.834608576486 byte
Avg Bytes yang di sent/sec	1031.325586707 byte
Avg Bytes dieksekusi / sec	1105.5623536801 byte
Current Bandwidth	2000000 bps
Total Bytes	38723427 byte
Service time per byte	1105.02063298 sec
Service Demand Incoming Link	0.000295338434306 sec
Service Demand Outgoing Link	0.00412530234683 sec
Utilization Incoming Link	0.326515054563 %
Utilization Outgoing Link	4.5607789722 %

Gambar 4.19 Hasil perhitungan MMC untuk disk dan incoming/outgoing link

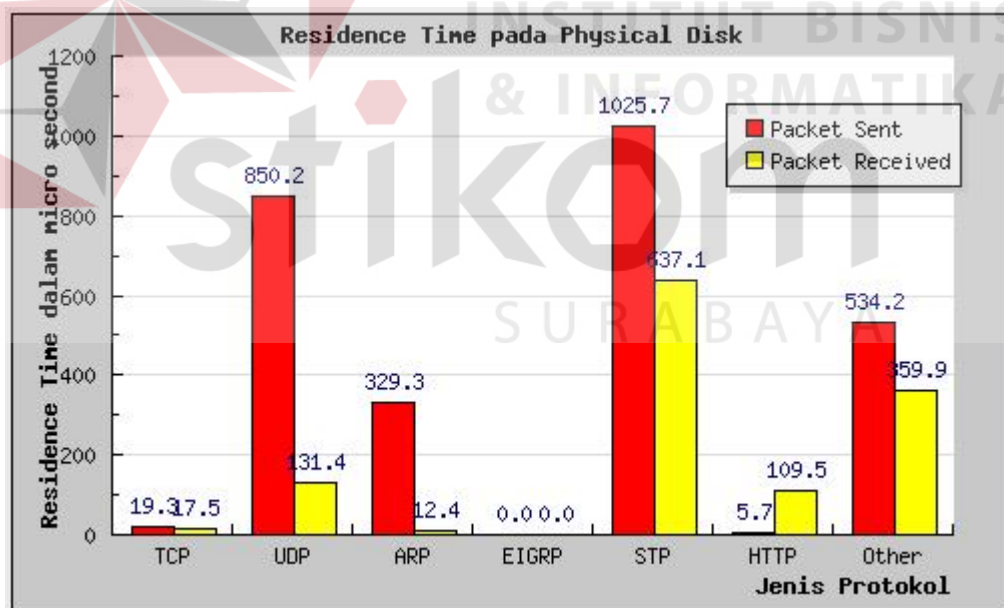
Menu berikutnya adalah Hasil Analisis per komponen per protokol. Analisis ini diperoleh dari dua perhitungan sebelumnya yaitu perhitungan per komponen dan perhitungan per protokol. Contoh hasil analisis tersebut ditunjukkan pada Gambar 4.20 dengan level komponen incoming link dan processor.

Resources(Protokol)	Service Demand(μ sec)	Utilization(%)	Residence Time(μ sec)
Processor	1.477173	2.908553	1.521424
TCP Sent	0.816374	1.607440	0.840830
TCP Received	0.743102	1.463168	0.765363
UDP Sent	36.018764	70.920943	37.097773
UDP Received	5.565643	10.958751	5.732372
ARP Sent	13.951992	27.471472	14.369950

Resources(Protokol)	Service Demand(μ sec)	Utilization(%)	Residence Time(μ sec)
ARP Received	0.526648	1.036971	0.542425
EIGRP Sent	0	0	0
EIGRP Received	0	0	0
STP Sent	43.451585	85.556167	44.753258
STP Received	26.989058	53.141452	27.797565
HTTP Sent	0.242498	0.477479	0.249763
HTTP Received	4.639701	9.135572	4.778692
Others Sent	22.629485	44.557454	23.307393
Others Received	15.245370	30.018132	15.702073

Gambar 4.20. Analisis per komponen per protokol

Analisis Kinerja Level Komponen dan Protokol Jaringan juga disajikan dengan grafik batang/histogram. Gambar 4.21 adalah grafik untuk perhitungan Residence Time pada level komponen Physical Disk.

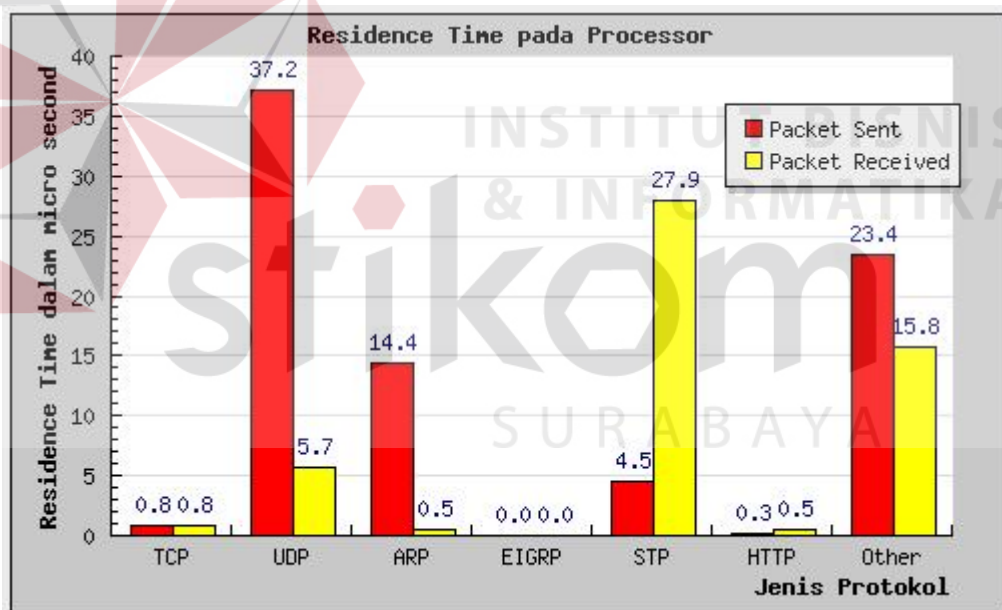


Gambar 4.21. Grafik residence time pada level komponen Physical Disk

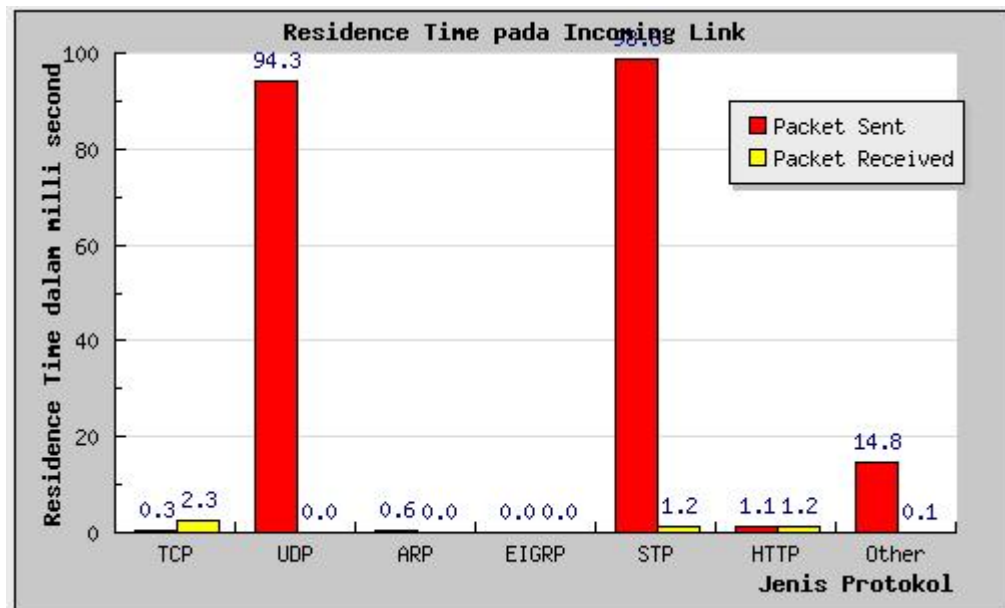
Dari Gambar 4.21, diketahui bahwa jenis protokol yang memerlukan waktu terlama untuk diolah oleh komponen Disk adalah Spanning Tree Protokol (STP), yaitu sebesar 1025,7 μ sec untuk frame yang masuk ke server dan 637,1

μ sec untuk frame yang keluar dari server. Jenis protokol yang memerlukan waktu tercepat diolah oleh komponen physical disk adalah TCP yang hampir mendekati nilai 20 μ sec untuk frame yang keluar dari server.

Pengukuran residence time pada processor/CPU seperti tampak pada Gambar 4.22, yang menunjukkan bahwa protokol UDP merupakan protokol yang paling lama diproses oleh komponen processor sebesar 37,2 μ sec untuk frame yang keluar dari server dan 5,7 μ sec untuk frame yang masuk ke server. Sedangkan untuk STP memerlukan waktu untuk memproses frame yang masuk ke server selama 27,9 μ sec dan 4,5 μ sec untuk frame yang keluar dari server.



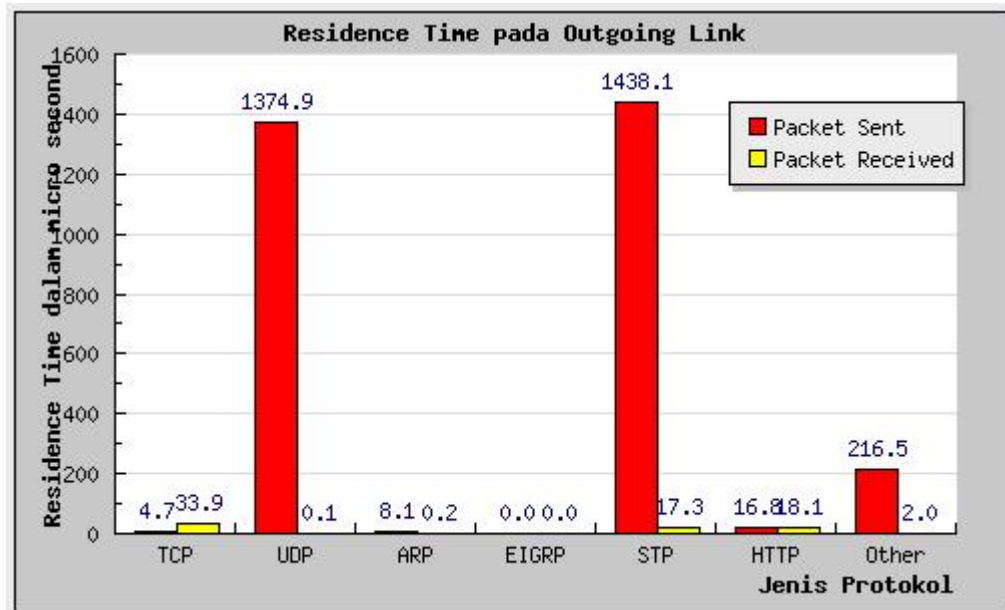
Gambar 4.22. Grafik residence time pada level komponen processor



Gambar 4.23 Grafik residence time pada level komponen incoming link

Gambar 4.23 adalah grafik residence time pada level komponen incoming link. Dari Gambar tersebut disimpulkan bahwa waktu yang diperlukan untuk memproses protokol yang paling lama adalah STP pada frame yang keluar dari server sebesar 98,5 μ sec dan 1,2 μ sec untuk frame yang masuk ke server. Pemrosesan terlama juga terjadi pada protokol UDP untuk frame yang keluar dari server selama 94,3 μ sec dan hampir 0,0 μ sec untuk frame yang masuk ke server.

Gambar 4.24 menampilkan grafik residence time pada level komponen outgoing link. Dari gambar tersebut dapat disimpulkan bahwa protokol STP memerlukan waktu rata-rata untuk diproses oleh komponen outgoing link sebesar 1438,1 μ sec untuk frame yang keluar dari server menyusul kemudian frame UDP yang keluar dari server selama 1374,9 μ sec.



Gambar 4.24 Grafik residence time pada komponen outgoing link

Dari serangkaian evaluasi yang dilakukan, dapat diketahui bahwa protokol STP merupakan jenis protokol yang paling lama diolah oleh semua level komponen kecuali processor.

Jenis resource yang memerlukan waktu tercepat untuk memproses protokol adalah pada level komponen processor. Sebagai contoh untuk memproses protokol STP, processor membutuhkan waktu 27,9 µsec, 637 µsec untuk level physical disk, 1,2 msec untuk level komponen incoming link dan 1438,1 µsec untuk level komponen outgoing link.

Dari hasil analisis yang dilakukan pada protokol, jumlah frame STP relatif sangat kecil dibandingkan dengan jumlah frame protokol HTTP, namun untuk melayani protokol STP ini, dibutuhkan waktu yang relatif lama dibandingkan dengan protokol HTTP. Jumlah bytes protokol STP relatif sangat kecil dibandingkan dengan jumlah bytes pada protokol lain, namun waktu untuk melayani protokol ini sangat lama.