CHAPTER IV

SCHEDULING SCHEME FOR MANAGING MEDIA TRAFFIC IN NETWORK
Me-TrffSchl - (Media Traffic Scheduler)

4.1 Introduction

Streaming real-time media requires consistent quality measures to be adopted during transmission and play-out at receiver end. Without providing consistent QoS, real-time traffic experiences jitter, variations in delay during play-out, and packet loss. The need for differentiated traffic scheduler (Me-TrffSchl) architecture to manage and smoothen the bursty media traffic, which should cater to delivery of quality of real time media streaming application.

Me-TrffSchl algorithm discusses on the efforts taken to manage multiple type of traffic streams in heavily congested network. The algorithm works by assigning rule-based priority to inflow packets, identifying weight-based optimal links and setting priority limits for media transfer in buffer. This algorithm has been developed to avoid media packet loss as well to minimize queuing delay. It has been identified in Chapter-III that end-to-end delay obtained in Me-PLM converges to delay of SF-BLUE. Effective management of variable traffic parameters with priority assigned on media packets is implemented at run-time. This prepares the gateway for a deterministic and adaptable scheduling scheme, which minimizes media packet loss and reduces end-to-end delay at receiving end. This scheme has introduced a novel idea of setting priority to media packets at run-time by marking them in their headers as well minimizes jitter by providing synchronization between packets.
4.2 Challenges in Media Network Traffic

The workload analysis [107] of a streaming conference application over IP network shows that client demands in media session highly vary at different time intervals. Additionally, media packets and traffic should be characterized by stringent real time constraints, as each stream possesses asynchronous play-out. Best - Effort [78] scheme considers that all packets are similar and provides with same set of properties, packets adopt same traffic speed, delay is added for media packets. These pitfalls have lead to the design of an adaptive multiple differentiated media scheduler architecture for traffic management to cater to needs of real time media streaming applications.

Smoothing of varying traffic, for delivery of variable bit-rate (VBR) media streams [93] is an important challenge in design of multimedia network protocols. Generally in client buffer, the window size and sliding-window size should be increased to allocate packets of varying size. Previous window-based online smoothing methods [93] tried to reduce peak bandwidth allocated to each window. However, as bandwidths allocated in different windows are handled independently, which requires large traffic for transmitting entire stream.

Table 4.1 shows traffic generated, and buffer requirement for various network applications. It is found that applications like web browsing, e-mail, and text based chat require less buffer and bandwidth for transfer. Applications like media conferencing, media-on-demand, which transfer large chunks of media packets through the congested network, require high buffers [88] for play-out to avoid delay and jitter. It also demands high bandwidth for managing loss of packets when transferred through network. Managing traffic
Table-4.1 Traffic behavior of network applications

<table>
<thead>
<tr>
<th>Applications</th>
<th>Traffic Behaviour</th>
<th>QoS Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail Transfer (E-Mail)</td>
<td>Small Batch File</td>
<td>Tolerant of Delay</td>
</tr>
<tr>
<td>File Transfer (FTP)</td>
<td>Transfer</td>
<td>Bandwidth requirement – Low</td>
</tr>
<tr>
<td>Remote Terminal</td>
<td></td>
<td>Best Effort</td>
</tr>
<tr>
<td>HTML Web Browsing</td>
<td>Bursty File Transfer</td>
<td>Tolerant of Moderate Delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bandwidth requirement - Varies</td>
</tr>
<tr>
<td>Client – Server</td>
<td>Small two way</td>
<td>Sensitive to loss &amp; delay</td>
</tr>
<tr>
<td>E- Commerce</td>
<td>transactions</td>
<td>Bandwidth requirement- Moderate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be reliable</td>
</tr>
<tr>
<td>IP – Telephony</td>
<td>CBR or VBR</td>
<td>Sensitive to delay, jitter &amp; loss</td>
</tr>
<tr>
<td>Real Audio</td>
<td></td>
<td>Bandwidth requirement - High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires predictable delay &amp; loss.</td>
</tr>
<tr>
<td>Video Conference</td>
<td>VBR</td>
<td>Sensitive to delay, jitter &amp; loss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bandwidth requirement - High</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires predictable delay &amp; loss.</td>
</tr>
</tbody>
</table>

is the primary requirement to provide QoS in streaming media applications. Delay in media streaming networks has been a prominent issue, which degrades the quality of applications [88].

Play-out delay and end-to-end delay are primary issues to be discussed. It has been identified that video frames should be played within 100ms [39], while an audio frame should be played within a maximum time of 120ms[16]. The existing methods had received packets within an average delay time of 220ms, which is irreparable to be perceived by any user.

An average of 3.07% of packet loss has been achieved in DiffServ [13] and 4.87% in IntServ [90], which has acceptable latency effect. To achieve the best latency effect, packet loss should be negligible with an average of less than 2%. This section deals in detail on different challengeable parameters to achieve QoS.

4.2.1 Traffic Management and Classification

Network traffic needs to be identified and classified based on its behavior. Network traffic can be defined as constant bit rate (CBR), variable
bit rate (VBR) or bursty and adaptive bit rate (ABR), which is based on bit-rate incoming, while intensity defines nature of traffic. Counters can be used to identify the rate of packet arrival. An analysis [88] shows that real time traffic being variable in size of fluctuations have large varying amplitudes over long scales. The study provides an insight into the traffic pattern generation complexity, which leads to network causalities such as packet loss, collision, delay and other latency effects. These latency effects define the quality of media stream received at end terminals.

Subjective studies [31,121] have been conducted to access the impact of these effects. The network key parameters to be quantified for a media call are queue-blocking probabilities, which can manage end-to-end packet delay, inter-packet mean arrival time, end-to-end packet loss probability, header overhead, and throughput. The common goal of RED and SF-BLUE is to keep the queue size small in routers which functions by detecting queue buildup and notifying end-hosts before the queue in router overflows. With ECN, a router can signal congestion to an end-system by "marking" a packet. (Setting a bit in header). RED scheme controls traffic congestion by marking packets when packet drop out but efficient queue management does not result in better output.

Neither RTP nor RTCP provide mechanisms to ensure timely delivery or QoS guarantees or take care of the underlying network parameters such as traffic or backbone topology. Traffic can be controlled at network edges [18], internal points of network [112] and terminal network points. The method of classification provides simplicity where various traffic types use differentiated approach. Methods that classify at the edge of network such as traffic shapers, bandwidth managers apply the traffic rules at the entry points
(ingress nodes) and outgoing points (egress nodes) where traffic mostly passes. Traffic classification in intermediate points of network is done at routers, gateways, where classification is based on flow rates per connection, queuing conditions and packet sizes. End users and applications assign traffic classification at end-points. This method of classification is rarely trusted, since all users and applications prefer premium handling, which is impossible to attain.

Traffic being classified at edges and internal routers provide a converged differentiated services architecture. Me-TrffSchl uses both edge classification at buffers and internal traffic classification working at routers. The edge based traffic classification classifies the packets and marks packets based on packet type and transport protocols assigned. This scheme uses traffic handling methods after packet classification. Traffic Handling is being carried out at edge routers or ingress gateways where packets are serviced and sent to next gateway. DiffServ uses traffic classification on priority basis at edges. Here DiffServ's Assured Flow and Expedited Flow code points can be used to mark streaming and high priority traffic.

The assured probability measure of delay for play-out at end-hosts is due to the consequence of media packets being discarded as late arrival or re-transmission of dropped packets. In buffer oriented schedulers the continuity quality of the stream is affected by under-flow discontinuities (when buffer empties) and over-flow discontinuities (when buffer is full).

4.2.2 Delay Minimization

Minimization of delay is characterized by stringent real time constraints, as each stream possess asynchronous play-out. There has been extensive work on measurements and characterization of delay and loss in Internet.
Bolot et al [16], sent audio traffic and measured delay and packet loss incurred. The delay invariability was found to have form of spikes where the graph shows mostly isolated packet loss as shown in Fig-4.1. From multicast measurements [17], the nature and characteristics of temporal and spatial correlation of packet loss were studied on WAN network. Markopoulou [4] conducted an experiment on WAN network using Best-Effort and High-Priority methods. From results, it was found that an average queuing delay of 150ms to 170ms was identified, where the cause is attributed to waiting at gateways. RED exhibits an average delay of 260ms for media applications, while 150ms for other applications. IntServ and DiffServ show an average delay of 190ms for media applications and SF-BLUE shows an average delay of 210ms. From the results it was noticed that the major part of the delay variance attributes to propagation and queuing process.

![Fig-4.1 Isolated Packet Loss](image)

4.3 Proposed Design - Media Traffic Scheduler (Me-TrffSchl)

The proposed Me-TrffSchl (Media Traffic scheduler) works on managing traffic by allotting buffer resources and marking priority on packets.
This scheme handles the incoming stream of packets to Packet-Handler and set priority to stream of media packets by marking IP packets. Traffic is smoothened by identifying types in Traffic-Extractor module and assigns into Queue-Controller module. Setting run-time limits over the active sorted queue controls the flow of media traffic effectively to increase the throughput of media transfer over highly congested broadband networks. This approach reduces media packet loss to an average of 1.15% during network bursty conditions. The design approach uses three layered functional modules to deliver effective QoS as Packet-Handler, Traffic-Extractor, Queue-Controller.

4.3.1 Classifying and Marking Packets

Media packets are classified and marked based on payload type of media application, packet size and transport protocol used for transmission. Payload media is categorized into audio, video and animated images. Information of payload is provided by application device at edge-terminals. STREAM_CONTEXT defines the payload type and media protocol in use.

```plaintext
Procedure Packet_Classification
Begin
If ((STREAM_CONTEXT = RTP) || (STREAM_CONTEXT = RTCP)) Then
  MEDIA_CONTEXT = RT_MEDIA
  // Definitions media context for video and audio
Else If (STREAM_CONTEXT = RTP) Then
  MEDIA_CONTEXT = OL_MEDIA
  // Off Line Audio
Else (STREAM CONTEXT != RTP)
  MEDIA_CONTEXT = WEB_DATA
  // Web Data
End If
End.
```

Algorithm : Packet Classification

Priority is assigned to packets with MEDIA_CONTEXT with RT_MEDIA, whose payload holds real-time media packets. Least priority is
assigned to WEB_DATA, which holds text and character contents for e-mail. Packets are also classified based on EPT field and SYNC field.

4.3.2 Classifying Traffic

The transfer of voice over packet switched networks has been studied thoroughly in literature. Studies in these networks can be classified based on traffic service class architecture, which is mainly differentiated into Variable Bit Rate, Adaptive Bit Rate, and Constant Bit Rate streams. Me-TrffSchl concentrates on real time media streaming, traffic stream to be Variable Bit Rate and bursty. Real time application works on this type of traffic, such as real time voice, video or other types of media. Since VBR employs multiplexing, there exist possibility of packet loss. The study performed by Rajan [90] shows that maintaining the VBR media traffic pay-load to 50% of available link flow capacity results in good voice quality.

```
Procedure Traffic_Classification
Begin
    If (MEDIA_CONTEXT=RT_MEDIA) and (INT_ARRIVAL > 10) Then
        STREAM_TRAFFIC = IT_STREAM // Duration is short
        TRFF_PRIORITY = 1
    Else If (MEDIA_CONTEXT=RT_MEDIA) and (INT_ARRIVAL > 50 AND < 150)
        Then STREAM_TRAFFIC = BRT_STREAM // Bursty Traffic
        TRFF_PRIORITY = 2
    Else If (MEDIA_CONTEXT=OLMEDIA) and (INT_ARRIVAL > 100 AND < 200)
        Then STREAM_TRAFFIC = VB_STREAM
        TRFF_PRIORITY = 3
    Else If (MEDIA_CONTEXT=OL_MEDIA) and (INT_ARRIVAL < 100) Then
        TRFF_PRIORITY = 4
    Else If (MEDIA_CONTEXT=WEB_DATA) Then
        STREAM_TRAFFIC = CB_STREAM
        TRFF_PRIORITY = 5
End
```

Algorithm : Traffic Classification
Nikolaos and Ioannis [55] had identified that real-time objects such as streaming video clips store only a portion of the clip for play-out. Since storing a large lengthy video clip can add up to huge memory, and large delays for play-out. Hence generally in video playing systems temporary buffer is recommended which works like a cache to store fragments of video clips and play-out series of clips. But this method adds up to cost as well proper sequence has to be followed.

**INT.ARRIVAL** defines inter-arrival mean packet time between set of packets. mean difference between set of packets received.

**MEDIA_CONTEXT** defines the payload of media packet

**STREAM_TRAFFIC** specifies the traffic type of stream

**TRFF_PRIORITY** specifies the priority assigned to traffic

### 4.3.3 Service Based Queue (SBQ) Handler

Queue Handler learns the queue strength (Qst) or average queue length, before a packet has to be allotted into queue. ‘Qst’ is a variable parameter, which decides on the type and priority of packet, to be assigned into available queue. The allocation of a packet into queue is based on Markovian Birth-Death Model, since allocation of a packet into queue or removal of a packet from queue is always found to be exponential. Any increase in media packet on a path lead to increase in demand for bandwidth which allocates buffer from neighboring routers / gateways or freeing the idle buffer.
Fig-4.2 Queue Priority based on traffic and packet

Fig-4.2 shows multiple queue-based routers, which has multiple incoming links. Various types of packets are found in queue, which are controlled and assigned into buffer for out-going links based on Queue-Controller weight “MeQ”. This module assigns the packets into buffer on basis of packet type and traffic type.

Table-4.2 Buffer Parameters used in simulation

<table>
<thead>
<tr>
<th>Buffer parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBR</td>
<td>Total buffer space on request from source</td>
</tr>
<tr>
<td>CAB</td>
<td>Currently available buffer space</td>
</tr>
<tr>
<td>ABD</td>
<td>Adaptive bandwidth on demand</td>
</tr>
<tr>
<td>CRB</td>
<td>Connection bandwidth on request</td>
</tr>
<tr>
<td>$\mu_b$</td>
<td>Ideal buffer utilization point</td>
</tr>
<tr>
<td>$\delta_u$</td>
<td>Upper - bound buffer utilization limit</td>
</tr>
<tr>
<td>$\delta_l$</td>
<td>Lower - bound buffer utilization limit</td>
</tr>
</tbody>
</table>

Always buffer utilization limit varies $0 < \delta_l \leq \delta_u < 1$

Table-4.3 Traffic Intensity parameters

<table>
<thead>
<tr>
<th>Traffic parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_c$</td>
<td>Current instance of Traffic rate control flow of media stream.</td>
</tr>
<tr>
<td>$\tau_i$</td>
<td>Minimum acceptable flow</td>
</tr>
<tr>
<td>$\tau_a$</td>
<td>Maximum acceptable flow</td>
</tr>
<tr>
<td>$K_p$</td>
<td>Inter-arrival rate of incoming packets</td>
</tr>
<tr>
<td>$S_p$</td>
<td>Required Servicing rate of media packets</td>
</tr>
<tr>
<td>$K_b$</td>
<td>Availability of buffer at a time interval</td>
</tr>
<tr>
<td>$S_b$</td>
<td>Buffer size occupied within an interval of time</td>
</tr>
</tbody>
</table>

Always traffic utilization limit lies between $\tau_i < \tau_c > \tau_a$
Table-4.2 and Table-4.3 provides the buffer parameters and traffic parameters, which are used for traffic flow control.

**Procedure Flow_Rate_Control**

**Description:** For any incoming flow of packets, the total buffer space required is TBR. Traffic flow is controlled on basis of incoming rate of media packets, if incoming flow of packets is at rate of $\tau_c$ where $\tau_c < \mu_b$, then $\tau_i < \mu_b < \tau_a$.

**Begin**

[1] If $\text{CAB} \geq \text{TBR}$ then Accept_Session ( )

{Traffic stream rate control $\tau$ is found to be within flow limit $\mu$ for any streaming packet limit $i$. Hence $\tau < \mu_i$ where $i$ varies within $\delta_i < \delta_u$}

$\text{CRB = CAB - TBR}$, CRB is assigned buffer varying between $\delta_i$ and $\delta_u$

[2] If $\text{CAB} < \text{TBR}$ then Accept_Session ( )

{If $\tau < \delta_j$, such that the required flow rate $\mu$ can be managed. Traffic stream rate control $\tau$ is found to be within minimum utilization limit $\delta_j$ while buffer utilization limit $\mu$ for any streaming packet limit $i$. Hence $\tau < \mu_i$ where $i$ vary within $(\delta_i, \delta_u)$. Increase in media payload results in increase in buffer by which delay time also increases which may be consistent.}

[3] Instance of Traffic Flow $\tau_c$ defines the tolerance limit of traffic flow through network, for which the traffic can be assigned for a link or path of network at any instant of time. $\tau_c = 2 \cdot \log e \left( \frac{K_p}{S_p} \right)$

{The Instance of traffic flow ($\tau_c$) invariably depends on Buffer Utilization point ($\mu_b$), as well depends on rate of flow of both incoming and outgoing media packets such as inter-arrival rate of media packets ($K_p$) and service rate of each packet ($S_p$)}. Here,

$\tau_i$, the minimum acceptable flow $= \log e^{\tau_c}$
Buffer Utilization Point $\mu_b$ defines the tolerance utilization point, which can be achieved for the buffer resource available at hand.

$$\mu_b = 2^* \log_e \left( \frac{K_b}{S_b} \right)$$

{The buffer utilization factor limits on availability of buffer at a time interval $(K_b)$ and buffer size occupied within an interval of time $(S_b)$.} End.

Algorithm: Managing Traffic Flow

Strict simulation parameters provide an effective solution for the best case of network traffic or worst case of bursty traffic when buffer is not sufficient to transfer the packets. The algorithm identifies an optimal out-going link when the traffic is worst or if enough buffers are not available.

4.3.4 Early Play-out Time (EPT)

RTP provides high priority to media packets, but proper control over traffic is not discussed. Media Packets are transmitted with primary field EPT that assigns highest priority to media packets during transfer. The aim of setting EPT enables media packets to be transferred within the specified time limit. All media packets marked with EPT are to be serviced under higher priority at each router, so that they may reach the end-host in time, thus avoiding delay. Video packets carry an average EPT of 100ms, which is acceptable to achieve the required quality. A real time audio packet requires an average of 120ms play out time while an off-line media packet requires an average EPT to be 135ms. Media packets are marked with EPT field in Me-TrffSchl header.

EPT is similar to TTL (Time To Live), which specifies time limit for the packet to live in the network, and when the limit is crossed, packet is dropped.
by which real time consistency is maintained. EPT differs from setting TTL, where packets are not dropped but identified and accelerated at each router based on flow intensity. EPT value is decreased in milliseconds at each gateway, for the time packet resides. Hence at each gateway the time taken by packet to reside in gateway is reduced from actual EPT value and rewritten. If EPT reaches zero (0x000), then such media packets are considered as "late packets" and checked for synchronization by router. The synchronized late packets provide information about the link through which packet has traveled, traffic intensity in path, and buffer parameters which decide on acceptance of media packet or dropping. EPT is represented as in Fig-4.3. Media packets, which have crossed the specified EPT starts incrementing from next router, such that the lapsed time is obtained.

<table>
<thead>
<tr>
<th>EPT</th>
<th>SYNC</th>
<th>Stream Context</th>
<th>Stream Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9</td>
<td>16</td>
<td>24</td>
</tr>
</tbody>
</table>

Fig-4.3 Me-TrffSchl header format

Header Structure is specified as in Fig-4.3. It consists of the following fields:
- EPT – Specifies the early play-out time of media stream in hexadecimal
- SYNC – Hexadecimal value represents Boolean value of synchronization bit
- Stream Context – Specifies the media stream type.
- Stream Traffic – Specifies the traffic type of stream

EPT values can be configured for media packets as in Table-4.4. In 8-bits, only first six bits are used the remaining 2-bits are reserved for future. Table explains the EPT value for corresponding MEDIA_CONTEXT and STREAM_CONTEXT.

<table>
<thead>
<tr>
<th>MEDIA_CONTEXT</th>
<th>STREAM_CONTEXT</th>
<th>EPT Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT_MEDIA</td>
<td>IT_STREAM</td>
<td>010010101</td>
</tr>
<tr>
<td>RT_MEDIA</td>
<td>BT_STREAM</td>
<td>100010000</td>
</tr>
<tr>
<td>OL_MEDIA</td>
<td>VB_STREAM</td>
<td>100100101</td>
</tr>
</tbody>
</table>
4.3.5 **Synchronization of Packets**

Synchronization is the method of fixing media packets to be delivered within specified limits. Each media packet is transmitted with a SYNC bit field, which is used to identify whether an incoming media packet in buffer is within synchronization limit. SYNC field in media packet checks for SYNC field in each preceding and succeeding media packet. If the Sequence number of each packet is found in required order and well within EPT limit, then SYNC bit is set. If packets are not found in sequence number order higher than acceptable error of $10^9$, and not within EPT limit, then SYNC bit is reset. SYNC field is represented as in Fig-4.3.

Each packet is transmitted through a set of routers and gateways before reaching the receiver. Packets are marked with sequence number (Seq-No) as in IP header. Seq-No in each packet provides synchronization with next set of packets. If receiver time is 30ms above the EPT value, then packet is dropped. If packets are received above specified EPT and within 30ms of time, then packets are accepted.

### 4.4 Me-TrffSchl Architecture

![Fig-4.4 Me-TrffSchl Architecture](image-url)

- **Packet Handler**
- **Traffic Extractor**
- **Queue Controller**
- **Variable Bit Rate**
- **Early Play-out Time**
- **Traffic Extractor**
- **Apply Service Level Policy Architecture**
- **Assign Packets**
- **To next Gateway**
Me-TrffSchl architecture is depicted as in Fig-4.4, which consists of three layers, Packet Handler, Traffic Extractor, and Queue Controller. Each incoming packets are identified and classified by Packet Handler, and media packets are marked with EPT and SYNC fields.

Packets with high EPT and SYNC bit set are given higher priority to enter into Traffic Extractor module. This module identifies the traffic intensity in each link, and assigns weight for each link (MeQ). Assigning weight to link depends on buffer availability and average length of packets waiting in queue (Qst). The weight of queue "MeQ", priority of packets and priority of traffic intensity determine the number packets to be assigned.

```plaintext
Procedure Queue_Controller
Begin
  If (MEDIA_CONTEXT = RT_MEDIA) and (STREAM_TRAFFIC = IT_STREAM) and Qst = LOW then
    MeQ = 1 // Allot media packet into priority link '1'
    Assign_Path()
  Else
    If (MEDIA_CONTEXT = RT_MEDIA) and (STREAM_TRAFFIC = BRT_STREAM) and Qst = LOW then
      MeQ = 2 // Allot packet into priority link '2'
      Assign_Path()
    Else
      If (MEDIA_CONTEXT = RT_MEDIA) and (STREAM_TRAFFIC = BRT_STREAM) and Qst = HIGH then
        MeQ = 0 // Don't allot packets
        Alter_Path()
      Else
        If (MEDIA_CONTEXT = OL_MEDIA) and (STREAM_TRAFFIC = BRT_STREAM) and Qst = NORMAL then
          MeQ = 3 // Allot packet into high priority link '3'
          Assign_Path()
        Else
          If (MEDIA_CONTEXT = OL_MEDIA) and (STREAM_TRAFFIC = VB_STREAM) and Qst = NORMAL then
            MeQ = 3 // Allot media packet into high priority link
            Assign_Path()
          Else
            If (MEDIA_CONTEXT = WEB_DATA) and (STREAM_TRAFFIC = VB_STREAM) and Qst = LOW then
              MeQ = 4
              Assign_Path()
            Else
              If (MEDIA_CONTEXT = OL_MEDIA) and (STREAM_TRAFFIC = IT_STREAM) and Qst = LOW then
                MeQ = 1
                Assign_Path()
              Else
                If (MEDIA_CONTEXT = RT_MEDIA) and (STREAM_TRAFFIC = IT_STREAM) and Qst = LOW then
                  MeQ = 1
                  Assign_Path()
                End If
              End if
            End if
          End if
        End if
      End if
    End if
  End if
End if
```

Algorithm : Queue controller for packet allocation
Function Assign_Path() assigns the required path for packets to be allocated. Function Alter_Path() identifies another alternate path for allocating media packet. If buffer assigned is not available then Alter_Path() function is called, to allocate another path with required buffer. In condition of all paths strength being HIGH (i.e., Qst = HIGH) then non-media packets and less priority packets should be dropped to allocate high priority media packets.

4.4.1 Implementation

MeTrffSchl can be tested on implementing the algorithm in gateway server or router of network domain. Variable or bursty traffic propagation is based on recent large-scale analysis of media traffic [3]. This model carries an application-level description of critical elements and required components, which decide on the effectiveness of providing QoS. Scheme uses critical elements that characterize web centric protocols HTTP/1.0 and HTTP/1.1 protocols which manage the key request and reply session as well media centric session protocols such as SIP and H323. Other protocol components used to identify media traffic are RTP, RTCP for real time media packets. Real time streaming data is identified streaming protocol (ST) [6] such as RTSP or ST2 [76], which is used to stream voice, video and other real time transmission data is used to stream data. The offered media load is defined as network traffic resulting from emulating browsing behavior of a fixed-size population of users or type of media applications activated by different users.

The algorithm identifies the number of packets incoming for various intervals of time. The router identifies the buffer available and required buffer allocating the packets. The out-going links from the router and their
corresponding queue capacity, the traffic intensity available in router paths at any interval of time can be gathered from network software modules. MeQ (weight of link) can be set for each links and packets can be assigned. The experimental test methods and their performance analysis are discussed while preparing the test method is shown in Appendix-B.

4.5 Experimental Approach and Simulation

Experimental test-bed is prepared to analyze and verify Me-TrffSchl scheme. Two experimental test methods are adopted, simulator based experimental approach and a video conference experiment, which is conducted between four different locations. Both the experiments are used to identify the media packet loss, play-out delay, and link capacity for each scheme. The test bed adopted uses compressed media sources in a rate-controlled network. The system transmits encoded, compressed media (a combination of both audio and video frames) through the IP network. PCM G.711 is the compression method and coding method followed, which provides 64kbps compression of media data into IP network.

4.5.1 Experimental Setup-1

Experiment was conducted after initializing and configuring all router and end-system parameters such as buffer constraints, traffic intensity and queue length. Fig-4.5 shows test bed configuration consisting of three different networks Birds, Fighter and Mammals, which are interconnected through each other using two routers R1, R2 to eliminate congestion. The traffic intensity created between R1 and R2 is 50Mbps. Multiple instances of video clips were run at each client location to create aggregate traffic of 50Mbps. Client systems are configured with media centric applications such
as Video Play-out for multiple active users. C-coded Me-TrffSchl scheme is installed at each server. Table-4.6 shows the values gathered for videoconference conducted at four locations.

![Network Model Diagram](image)

The Network Model used in our simulation method is WAN interconnection setup where sender is Fighter network server at MAT Systems, Sunnyvale (US) transmits media data through network which is received by Birds network server at MAT Systems, Coimbatore, India (Receiver). Strict simulation parameters provide an effective solution for the best case of network traffic or worst case of bursty traffic when bandwidth is not sufficient to transfer the packets. The algorithm identifies an optimal solution when the traffic is worst or if enough bandwidth is not available.

### 4.5.2 Results and Discussion

Video Play-out software WinAmp [122] was used for this purpose, which enables a maximum of 10 users to connect online for discussion on minimum required bandwidth of 50Mbps at receiver for media packet play-out. Hence the experiment was carried out over two different Ethernet networks interconnected to Internet through multiple routers and gateways. An active
monitoring system is maintained at the router, to gather the media packet information and forward it to the server of corresponding client systems. Various network packet transfer parameters and networking latency factors are determined (shown in Table-4.5).

Test 1: Delay Calculation

Fig-4.6 shows the number of media packets received within the specified delay time for a set of sources (video clips) transferred over links of 100Mbps of Ethernet-LAN. The packets received for a single source of playing movie clip “The Last Emperor” from US domain network “Fighter” was received by “Birds” domain in India (Fig-4.5). The result shows the performance of the Me-TrffSchl algorithm, where the time for end-to-end play-out being received by end-host is found to be varying by an average of 110 ms to 135 ms, while DiffServ exhibits an average delay of 130ms to 180ms. Since video packets were scheduled with EPT field and SYNC, where EPT for video stream is 100ms, the scheme also shows an effective jitter removal. The queue buffer is maintained between 80Kbps to 120Kpbs, which is desired criteria for avoiding congestion, which is 80% of available queue capacity.

Test2: Link Utilization

The algorithm produces the best effort result as the time taken to receive the media packets over network whose bandwidth is variable depending on number of incoming media packets at an interval of time. Fig-4.6 defines the optimal link bandwidth occupied by network for various increase in source rate. The performance graph displays the change in source rate as a result of more contention of bandwidth.

The graph in Fig-4.7 shows optimal link utilization of Me-TrffSchl scheme compared with link utilization of DiffServ Scheme and Me-PLM
Scheme. Me-TrffSchl occupies an average link of 57Kbps, while Me-PLM requires 80Kbps and DiffServ occupies 60Kbps. Harmut and Thorsten confirm the result in their work [34].

Test 3: Packet Loss

Fig-4.8 shows the percentile loss of media packets for video play-out application, run over two geographically different locations. I, II, III, IV denote the sources (set of video clips), which were run for 30minutes. Me-TrffSchl scheme exhibits media packet loss of 1.05% to 1.10% compared to DiffServ.

4.5.3 Experimental Setup-2

Worst-case situation of admitting bursty variable media traffic into the network is the primary concern. The best-case situation can be identified for network, which admits variable media packets whose flow is constant or allocated for reserved buffer. Real time data gathered from Internet on Best-Effort Scheme are populated into Me-TrffSchl Scheme simulated using ns2 simulator [70], which identifies the traffic free optimal link to transfer the media packets at each intermediate gateway. Monitoring module residing at each gateway or router monitors the incoming media packets and each outgoing media packets in network as well into adjacent domain through egress routers. Various simulation parameters used are shown in Table-4.5.

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi</td>
<td>Number of Packets</td>
</tr>
<tr>
<td>Ts</td>
<td>Time of arrival at the gateway from In-flow</td>
</tr>
<tr>
<td>Tr</td>
<td>Time to transfer packet(s) from gateway T to out-flow j</td>
</tr>
<tr>
<td>Wt</td>
<td>Waiting time of packets at Gateway T</td>
</tr>
<tr>
<td>Oi</td>
<td>Route i of out-going path</td>
</tr>
<tr>
<td>li</td>
<td>Route i of in-coming path</td>
</tr>
<tr>
<td>Bi</td>
<td>Bandwidth of in-flow</td>
</tr>
<tr>
<td>Bo</td>
<td>Available Bandwidth of out-flow</td>
</tr>
<tr>
<td>Ps</td>
<td>Packet Size</td>
</tr>
<tr>
<td>Gc</td>
<td>Capacity of Gateway to Service</td>
</tr>
<tr>
<td>Pli</td>
<td>Percentage of Packets Lost</td>
</tr>
</tbody>
</table>
4.5.2.1 Results and Discussion

Test-1: Link Utilization
Fig-4.9 shows the link or bandwidth utilization for increase in number of users ns2 simulator test the link utilization for RED scheme, SF-BLUE, Me-PLM scheme and Me-TrffSchl scheme. Link utilization invariably depends on number of packets waiting at any instant of time in buffer for servicing. Waiting of packet in buffer adds to delay time and also increases packet loss. Me-TrffSchl shows an average of 55–60Kbps for a maximum of 100 users. DiffServ exhibits an average of 60-80Kbps, which is similar to Me-PLM scheme, which also shows an average of 65-80 Mbps for any set of packets.

Test-2: Play-out Delay
Fig-4.10 shows the play-out delay for varying media sources at different time intervals. The delay obtained by Best-Effort scheme and DiffServ scheme is compared with Me-TrffSchl scheme. It is found that Me-TrffSchl scheme shows an average delay of 110ms to 134ms, which is similar to video play-out experiment.

4.6 Summary
Me-TrffSchl scheme provides negligible media packet loss of less than 1.1%, which is better than Me-PLM scheme. This scheme has exhibited an average end-to-end delay of 110ms–130ms comparatively reduced to other existing literature based schemes. The percentage of media packet loss is found to varying from 1.07% to 1.16%, which is comparatively less than DiffServ scheme. The complexity of the algorithmic scheme is implementation of extended parameters such as EPT and SYNC and need for efficient control over sensitive traffic streams is stressed.
<table>
<thead>
<tr>
<th>Source-From</th>
<th>Source-To</th>
<th>% Packet Loss</th>
<th>Average Delay (ms)</th>
<th>% Packet Loss</th>
<th>Average Delay (ms)</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>D</td>
<td>5</td>
<td>145</td>
<td>15</td>
<td>174</td>
<td>00.30.05</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>3</td>
<td>132</td>
<td>11</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>5</td>
<td>139</td>
<td>13</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>B</td>
<td>6</td>
<td>147</td>
<td>15</td>
<td>162</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>1</td>
<td>99</td>
<td>7</td>
<td>104</td>
<td>01.10.04</td>
</tr>
<tr>
<td>B</td>
<td>D</td>
<td>1</td>
<td>139</td>
<td>4</td>
<td>167</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>1</td>
<td>133</td>
<td>4</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>B</td>
<td>0</td>
<td>121</td>
<td>3</td>
<td>157</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>0</td>
<td>125</td>
<td>3</td>
<td>198</td>
<td>01.30.00</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>0</td>
<td>90</td>
<td>3</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>0</td>
<td>150</td>
<td>2</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>C</td>
<td>0</td>
<td>102</td>
<td>5</td>
<td>130</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>0</td>
<td>98</td>
<td>4</td>
<td>121</td>
<td>02.00.07</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>0</td>
<td>113</td>
<td>3</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>1</td>
<td>90</td>
<td>6</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>B</td>
<td>0</td>
<td>152</td>
<td>0</td>
<td>192</td>
<td></td>
</tr>
</tbody>
</table>

A - MAT Systems Inc, Denver
B - MAT Systems Inc, California
C - Hisco (I) Pvt, Bangalore
D - MAT Systems, Coimbatore
Fig. 4.6 Delay times (ms) for Video Play-out Experiment - 1.
Link Utilization

- Me-PLM
- DiffServ
- Me-TrfSchl

Fig-4.7 Link Capacity for Video Play-out

Source

Link (bps)
Fig-4.8 Media Packet Loss for Video Play-out
<table>
<thead>
<tr>
<th>Type of Media Source</th>
<th>Packets Sent (N/sec)</th>
<th>Packets Received (N/sec)</th>
<th>Number of Packets Lost (N/sec)</th>
<th>Delay Encountered (msec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1276</td>
<td>1247</td>
<td>29</td>
<td>110</td>
</tr>
<tr>
<td>AV</td>
<td>1345</td>
<td>1290</td>
<td>55</td>
<td>120</td>
</tr>
<tr>
<td>A</td>
<td>1200</td>
<td>1200</td>
<td>0</td>
<td>95</td>
</tr>
<tr>
<td>A</td>
<td>1107</td>
<td>1101</td>
<td>6</td>
<td>97</td>
</tr>
<tr>
<td>AV</td>
<td>1466</td>
<td>1378</td>
<td>88</td>
<td>180</td>
</tr>
<tr>
<td>A</td>
<td>1035</td>
<td>1011</td>
<td>24</td>
<td>140</td>
</tr>
<tr>
<td>V</td>
<td>1532</td>
<td>1459</td>
<td>73</td>
<td>170</td>
</tr>
<tr>
<td>V</td>
<td>1459</td>
<td>1354</td>
<td>105</td>
<td>178</td>
</tr>
<tr>
<td>V</td>
<td>1356</td>
<td>1321</td>
<td>35</td>
<td>140</td>
</tr>
<tr>
<td>AV</td>
<td>1452</td>
<td>1423</td>
<td>29</td>
<td>135</td>
</tr>
<tr>
<td>A</td>
<td>1109</td>
<td>1109</td>
<td>0</td>
<td>102</td>
</tr>
</tbody>
</table>
Fig-4.10 Delay Time – Experiment 2

Delay Calculation

- MeTrfSchl
- BestEffort
- DiffSrv

Time (ms) vs Media Packet Source

Fig-4.10 Delay Time – Experiment 2