Strategies for Efficient Streaming in *Delay-tolerant* Multimedia Applications

Saraswathi Krithivasan, Sridhar Iyer
IIT Bombay
saras,sri @it.iitb.ac.in

Abstract

We consider Multimedia applications where clients specify a minimum desired stream rate (quality) and a time till when they are willing to wait, termed ***delay tolerance****. The Content Service Provider's (CSP) objective is to service as many clients' requests for the same multimedia content with a single stream while satisfying their requirements. We propose an optimization approach to determine the rates delivered at clients and study three transcoder deployment strategies:*
(i) ***Source Transcoding (ST):*** *when content encoded at different rates are available only at the source*
*(ii)****Anywhere Transcoding (AT):*** *when transcoding capability is available at all intermediate nodes, and*
*(iii)****Selected Node Transcoding (SNT):*** *when transcoding capability is available at selected intermediate nodes.*
Considering the complexity of the optimal solution, we propose a set of heuristic based algorithms for delivering enhanced rates to clients using the three strategies. A practical content dissemination network is used to demonstrate the effectiveness of our strategies.

1. Introduction

With the proliferation of networks connecting different parts of the world, several popular streaming media applications have emerged including: universities offering their courses to a set of global subscribers, service providers streaming movies requested by their clients, and multinational corporations providing training to employees across cities. These multimedia applications are *delay-tolerant*, where clients request the start of play back at a convenient time specified by ($t+d_i$) where t is the current time and d_i is the *delay tolerance* acceptable to client C_i. Clients also specify a minimum acceptable quality, typically specified as the encoding rate. Typical characteristics of such applications include: (i) a *source* that is responsible for the dissemination of contents; (ii) a set of geographically distributed *clients* connected through links of varying capacities and characteristics, and (iii) intermediate nodes which forward the data termed *relay* nodes.

A review of the existing mechanisms (details are presented in Section 6.3) for effective and efficient delivery of multimedia in [4][5][12] indicates that existing work treats multimedia dissemination as a soft real-time application that can tolerate some transmission errors and aims to *minimize* the startup delay. In contrast, we focus on multimedia applications that *can* tolerate startup delays. We illustrate the problem using a simple example below:

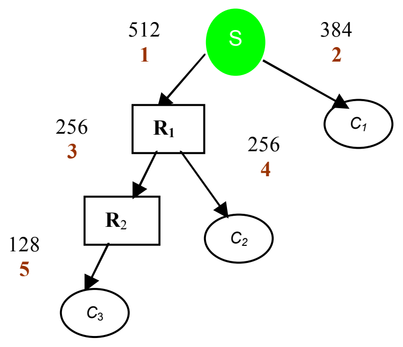


Figure 1: Motivating example

Source S is streaming contents to clients C_1, C_2, and C_3 connected with links 1 to 5, with bandwidths in kbps as indicated in Figure 1. Let the base encoding rate of the file be 512 kbps and the duration of the play out be 1 hour. We assume the same requirements for all three clients: a minimum rate of 128 kbps and delay tolerance of 30 minutes. Our objective is to provide the best possible rates to clients in this network. We consider the following cases:
(i) *Source Transcoding* (ST), where only the source is capable of transcoding, assuming zero delay tolerance: Clients require immediate play out without loss (delay tolerance =0). Weakest link bandwidths in the paths from S to C_1, C_2, and C_3 are 384 kbps, 256 kbps, and 128 kbps respectively. C_1 gets 384 kbps while both C_2 and C_3 get 128 kbps as they share the first link, and since no transcoding is possible in the relay nodes.
(ii) *Source Transcoding* (ST) considering clients' delay tolerance of ½ hr: The following rates can be delivered at the clients: C_1: 512 kbps, C_2: 384 kbps, and C_3: 192 kbps (refer to Section 2 for the expression used in the calculation). C_1 gets 512 kbps, being a client directly

connected to the source. However, in ST as no transcoding is possible in the relay nodes, both C_2 and C_3 get 192 kbps. Note that the delivered rates have improved for all the three clients when their delay tolerance is considered [7].

(iii) *Anywhere Transcoding* (AT), where transcoding capability is available at *all* relay nodes, considering clients' delay tolerance of ½ hr: C_1 still gets 512 kbps, the best possible rate, and C_2 gets 384 kbps as R_1 transcodes the stream from 512 kbps to 384 kbps for C_2. C_3 gets 192 kbps. However, when transcoders are available both at R_1 and R_2, there may be redundant use of transcoders to deliver 192 kbps to C_3. In this example two transcoders are used – R1 converts stream from 512 to 384 kbps, R2 from 384 to 192 kbps.

(iv) *Selected Node Transcoding* (SNT), where transcoding capability is provided only at selected nodes, considering clients' delay tolerance of ½ hr: By restricting transcoding capability only to R_1, we can still achieve the same rates at the clients as in case (iii) while reducing costs.

Note that the selection strategy can be based on several criteria, using simple logic all the way to exploiting complex information on the CSP's investment constraints and knowledge of client history. One example of a simple but effective selection rule that we use: select only those relay nodes that have more than one out going link.

From the above discussion, the following points emerge: (i) Clients' delay tolerance can be leveraged to provide enhanced rates to clients (ii) Transcoding capability is required at the relay nodes to serve the clients with enhanced rates without loss, and (iii) Through selective placement of transcoding capability at appropriate relay nodes, resources can be efficiently utilized by the CSP.

At the same time, from the CSP's perspective, the following questions remain:

1.Using a single stream and by exploiting the delay tolerance of the clients, what is the best rate that can be delivered to each client?

2. Assuming transcoding capability at (only some of) the nodes in the network, which transcoders have to be enabled and what rate conversions would they perform?

3. Delivered rates at clients depend on the network topology, link bandwidths, and client requirements. Given this, what transcoder placement strategy – ST, AT, or SNT– needs to be invoked such that resource utilization and revenue are maximized?

In this paper we address these questions using an optimization approach explained in Section 3. Given the complexity of the optimization problem we develop greedy heuristic based algorithms that approximate the optimal solution, presented in Section 4. We present a case study of a practical network (used for distance education) to understand the implication of using the proposed strategies in Section 5. Our conclusions are presented in Section 6.

2. Problem formulation

We consider a simple network having a source S, relay node R, and a client C as the basic block, as shown in Figure 2. (Table 1 defines the notation used in this section.)

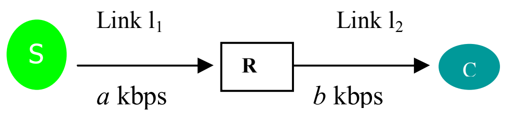


Figure 2: Basic block

Table 1: Notation used

α:	Base-encoding rate (highest fidelity) of the multimedia file at S.
r_c:	Stream rate delivered at C : Play out rate at C
d_c:	Delay tolerance specified by C
a:	Link bandwidth between S and R.: Link rate of l_1
b:	Link bandwidth between R and C.: Link rate of l_2
T:	Play out duration of the Multimedia content

Let $a < b$, i.e., Min $(a,b)= a$; Stream rate that can be delivered at C, r_c is given by:

$$r_c = \begin{cases} \alpha & \text{if } a >= \alpha\,; \\ a & \text{if } d_c=0 \text{ and } a < \alpha\,; \\ a + ((a^* d_c)/T) & \text{otherwise;} \end{cases} \qquad \textbf{(1)}$$

Equation (1) is derived as follows:

* When the bandwidth of the weakest link in C's path is greater than or equal to α, C can start the play out immediately at α, the best possible quality. If the client specifies a delay tolerance value, data needs to be buffered at the client for the duration of the delay tolerance.

* When the weakest link is less than α and the client's delay tolerance is zero, the delivered stream rate is equal to the weakest link rate for loss-free play out.

* Suppose C specifies a delay tolerance value d_c. *a* is the minimum bandwidth in the path between S and C. When the stream is encoded at *a* kbps, C receives it without any loss, immediately. But C waits for time d_c, its delay tolerance value before the play out starts. However, during this waiting time, an amount of data can be streamed to C is given by: $a * d_c$. The amount of extra data that C gets per second is: $a * d_c /T$, where T is the play out duration of the stream. Thus, the delivered stream rate at C is, $r_c == a + ((a^*d_c)/T)$.

Equation (1) provides the upper bound on the stream rate delivered at the client, as it considers the client's path in isolation. In a topology with multiple clients with shared links, finding the delivered stream rate at every client in the network is non-trivial. (Refer to Case (ii) of the motivating example). Also, given that our objective is to service clients without any loss, appropriate placement of transcoders in the relay nodes is necessary. In the following section we present an optimization approach to determine the client play-out rates for each of the three transcoder placement strategies: ST, AT, and SNT.

3. Optimization approach

We formulate our objective of maximizing delivered stream rates across all clients in the network as an optimization problem, where paths from source to every client in the network are considered:

Design variables -
Stream Rates x_m flowing through links l_m

Objective function -Maximize delivered stream rates at the clients, written as:
Minimize $\Sigma i\ (\alpha - x_i)^2$, where α is the base encoding rate and x_i is the stream rate flowing through link l_i. Since transcoding is a process where encoded rates can only be reduced, we desire to maximize the stream rates through every link in the network.

Constraints - Following constraints are applied to the objective function:

* *Rate constraint*: This constraint ensures that the stream rate flowing across any link in the network is bounded by the base-encoding rate, which is the highest possible stream rate. We represent this constraint as: $x_m <= \alpha$.

* *Transcoder constraint*: This constraint is used to respect the property of transcoding that incoming stream rate at the transcoding node is greater than or equal to its outgoing rate.

When ST policy is enforced, this constraint is specified as: for every client C_i, all link rates, x_i s, in its path are equal, represented as: ($x_s = x_n = \ldots = x_i$) for client i, having links l_s , $l_n,\ldots,l_i$ in its path.

Under AT, we specify the constraint as: $x_s >= x_n$; $x_m >= x_n$ .. for client i, having links l_s , l_m, l_n.. in its path.

In SNT, some relay nodes are selected as transcoding nodes. Suppose (l_s , l_a), (l_c, l_k) are pairs of links (incoming and outgoing) of nodes with transcoding capability and (l_b , l_m), (l_d, l_n) are pairs of links of nodes that do not have transcoding capability in the path of C_i. We use the inequality constraints ($x_s >= x_a$), ($x_c >= x_k$) and equality constraints ($x_b = x_m$), ($x_d = x_n$).

* *Delay tolerance constraint:* This constraint ensures that the client specified delay tolerance d_i is not exceeded while delivering enhanced rate to the client. It is specified as:

$L_i <= d_i$, where L_i is the latency incurred in the path from S to C_i, due to buffering and transcoding. We derive the expression for L_i below.

3.1. Expression for latency

Let L_i be the latency that will satisfy the delay tolerance specified by client C_i. There are two factors that contribute to L_i: (i) the latency introduced due to buffering L_b, when a stream encoded at a higher rate is flowing through a link having lesser bandwidth, and (ii) latency introduced due to transcoding in the relay nodes, L_t. As per the property of streaming, since the bits are streamlined, L_b depends on the bandwidth of the weakest link b_w, in the path from S to C_i. Let r_i be the delivered stream rate and T be the play out duration of the file. The expression for L_b is given by:

$$L_b = ((r_i - b_w)/ b_w)*T \quad \textbf{(2)}$$

To find the transcoding latency, we assume each transcoding enabled relay node to be a RTP/RTSP client as in [10]. As the bits are streamed, they go into an in-buffer (large enough to hold enough content required for transcoding) and the transcoded content is pushed into the out-buffer. The *processing rate* is defined as the number of bits processed by the transcoder per second. As long as this rate is more than *Min(bandwidth of the incoming link , bandwidth of outgoing link)*, there is no significant latency introduced due to the transcoding process. Also, compressed domain transcoding [8][11] has improved the speed of transcoding as it reuses motion information. Thus, we include a small *constant* transcoding latency L_t, the worst-case delay that may result from deploying transcoders in the relay nodes. Note that due to the real-time nature of the transcoding process and the pipelining property of streaming, irrespective of the number of transcoders used in the path of a client, transcoding latency remains constant. Thus, latency L_i is given by: $L_i = L_b + L_t$ **(3)**

3.2. Need for heuristics

We use fmincon function from the optimization toolbox of MATLAB to find the optimal delivered rates at the clients. Function fmincon uses Sequential Quadratic Programming (SQP) optimization method [9]. It is a gradient descent based search algorithm in the continuous search space. It starts from an initial point and usually converges to a constrained local optimum close to the initial point.

Considering our optimization formulation, suppose we have *m* links in the path, in the worst case the optimizer solves the problem by trying all possible values for the *n* distinct values corresponding to the effective rates that flow across the links. A total of m^n computations are needed. Thus, if there are 20 links in the path with each taking one of 15 possible values, the search space is of the order of 20^{15}.

To understand the complexity of the optimization algorithm, we generate 6 random topologies with randomly assigned link bandwidths having 20, 28, 36, 44, 52, and 60 nodes respectively. All other parameters including client requirements (minimum rate 128 kbps, Delay tolerance: ½ hr.) and transmission duration (1 hr.) are kept constant. Figure 3 presents the measured CPU time for the optimization algorithm to converge, when transcoding is allowed at all relay nodes (AT). We observe that when the number of nodes is increased three times, from 20 to 60, the CPU time taken for the algorithm to converge has increased by 16.85 times, from 0.42 seconds to 7.08 seconds.

Considering the complexity of the optimization algorithm, we present algorithms based on greedy heuristics in the next section.

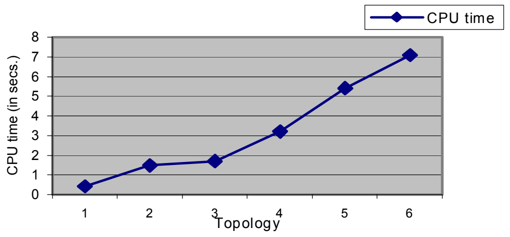


Figure 3: CPU time of optimization approach

4. Heuristic based approach

In this section we discuss algorithms for the three transcoder deployment strategies that can be used by the CSP that approximate the optimization algorithm. These algorithms are efficient; they converge within few seconds, even for topologies with hundreds of nodes, yet deliver good QoS to clients.

4.1. Local and global optimal rates

Algorithms presented below use equation (1) from Section 2 to find the best rate deliverable at each client, ignoring the constraints that arise due to the sharing of links in its path. This rate provides the upper bound on the rate deliverable at a given client. We term this *local optimal rate* for the client as the client is considered in isolation. We use the local optimal rates of all clients in a subtree to determine the rates delivered to clients in that subtree, termed *global optimal rates* as the entire subtree is considered. The algorithms are elaborated below.

4.2. Source Transcoding (ST)

In ST, the source is capable of providing different encoded rates for different subtrees emanating from it. Our goal is to service all clients in a subtree *without any loss* and *without using any transcoders at the relay nodes*. To achieve this goal, we consider each subtree rooted at the source and find the *minimum* of the deliverable local optimal rates in that subtree, considering the delay tolerance values of the clients. The source encodes the file at this rate for that subtree so that all clients in that subtree receive the multimedia content without loss. Referring to Figure 1 (Section 1), S transcodes the file to two versions at 512 kbps for C_1 in subtree 1 and 192 kbps for clients C_2 and C_3 in subtree 2. Detailed algorithm is presented in Table 2.

Table 2: Algorithm 1: ST

```
Input: currentroot: source id is given as currentroot
Output: globaloptimrates: rates delivered at clients

For each subtree rooted at currentroot
  If the subtree root is a client
    Find local optimal rate for the client %using Equation 1%
    Return value in globaloptimates for that client
  Else
    For each client in the subtree
      Find local optimal rate for the client %using Equation 1%
      Store value in localoptimrates()
    End
    Newrate= min(localoptimrates)
    For each client in the subtree
      Return newrate in globaloptimates
    End
  End
End
```

4.3. Selected Node Transcoding (SNT)

We present the algorithm for SNT first as AT is a generalization of SNT. Relay nodes that have more than one outgoing link are chosen to have transcoding capability, as any of these nodes are potential transcoding nodes.

The algorithm uses a greedy heuristic that executes the following steps:

(i) Starting from the source, for each subtree find the local optimal rates for clients in that subtree.

(ii) Find the **max** of these rates.

(iii) Assign this value to the link from the root of the tree to the subtree root (the first link in that subtree).

This process is recursively carried out for all subtrees at every level such that the stream rates flowing through each link in the tree is determined. Note that the stream rate flowing through the last link to client is the delivered rate at that client.

We use the **max** function to assign the stream rate to a link shared by many clients. Given our future research objectives where we consider dynamic client arrivals, this is an informed choice as transcoding is a one-way process, where bit rates can only be reduced.

With reference to Figure 1, link 2 in subtree 1 is assigned 512 kbps; As C_1 is the only client, delivered rate at C_1 is 512 kbps. Subtree 2 has two clients C_2 and C_3. Max of their local optimal rates is 384 kbps. This rate is assigned to link 1. The recursive algorithm now finds the subtrees rooted at R_1. The first subtree has only client C_2 and hence 384 kbps is assigned to link 4; C_2 gets 384 kbps. The second subtree has only client C_3, its local optimal rate being 192 kbps. This rate is assigned to link 3. Next step in the recursion considers the only subtree rooted at R_2 and assigns 192 kbps to link 5; C_3 gets 192 kbps. The top-level recursive algorithm is presented in Table 3.

Table 3: Algorithm 2: SNT

```
Inputs: currentroot: id of current subtree root
        currentrate:current stream rate
Output: globaloptimrates: rates delivered at clients

Globalalgo(currentroot,currentrate)
If currentroot is a client
   Compute localoptimrate %using equation 1%
   Return value in globaloptimates for that client
Else %if current root is not a client %
   Find the subtrees based at the currentroot
   For each subtree root
      If the subtree root is a client
           Find localoptimrate for the client %using equation 1%
           Store value in localoptimrates()
      Else
           For each client in the subtree
             Find localoptimrate for the client
             Store value in localoptimrates()
           End
      End
   End
   newrate= max(localoptimrates)
   If newrate is < previous link rate
           If currentroot is capable of transcoding % This
           additional condition is checked for SNT%
               Subtract transcoding delay from all affected clients'
               delay tolerance values and find their localoptimrates
             newrate= max(localoptimrates)
           End
     End
Call Globalalgo with subtreeroot as current root and newrate as
startrate
End
```

4.4. Anywhere Transcoding (AT)

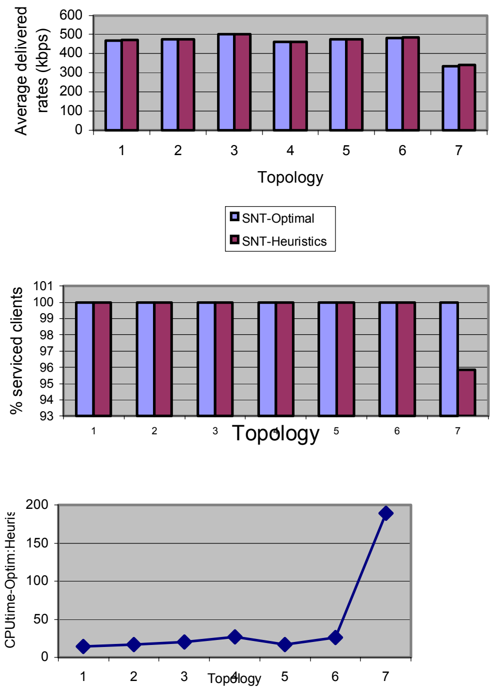
AT is the non-restrictive case of SNT. As all relay nodes are capable of transcoding, stream rates are assigned without the additional check in SNT (highlighted in SNT algorithm presented in Table 3). Considering Figure 1, the same steps are followed as in the case of SNT. For this simple example there is no difference in the stream rates assigned to links 1-5 for SNT and AT. However, note that in SNT only R_1 has the transcoding capability which is utilized. In AT, even though both R_1 and R_2 have the capability, only R_1's capability is utilized.

5. Performance evaluation

In this section, we present a comparative analysis of performance of our algorithms with the optimal algorithm. Following performance metrics are used:

(i) average rates delivered at the clients,
(ii) number of clients serviced, and
(iii) CPU time for convergence.

We have implemented the algorithms using MATLAB. We use the same six randomly generated topologies having 4 levels as discussed in Section 3.2. We randomly assign bandwidths chosen from 384 kbps to 1152 kbps to links up to level 3 and assign bandwidths randomly selected from 128 kbps to 384 kbps to the last links, as bottlenecks typically occur in the last mile. We also use as topology 7, a network with 200 nodes. We find that, for ST, the performance of our algorithm mimics the optimal algorithm *exactly* as our algorithm also finds the best possible rate delivered at clients using the weakest link in a sub-tree.



Figures 4(a)(b): SNT - Avg. delivered rate, % serviced clients; **4(c):** SNT -- CPU time ratio for optimal to heuristics

For AT and SNT, our algorithms work close to the optimal algorithms in predicting the delivered rates for clients. As explained in the previous section, our algorithms are based on greedy heuristics, where stream rates once assigned to links are not changed (no back tracking). Due to this, some clients may be denied service as client requirements may not be satisfied, especially in the case of SNT. This is seen in Figure 4(a) for topology 7. Figure 4(b) provides the average delivered rates at the clients. Note that for small topologies our algorithms work as well as the optimal algorithm. For a large topology as in topology 7, close to 96% of the clients are serviced; However, the average delivered rates predicted by our algorithms is better, as the average is calculated over fewer clients.

We plot the ratio of CPU time for optimal algorithm to our algorithm for SNT in Figure 4(c) that clearly indicates the applicability of our algorithms for quick decision-making in practical scenarios. In the next section, we present such a practical network scenario.

6. Case study: A distance education model

Our work was inspired by the Distance Education Program implemented at IIT Bombay [6]. Classroom lectures are synchronously transmitted to geographically distributed clients over a heterogeneous network (for example, the dissemination path to a client may include satellite and terrestrial lines). A client can subscribe to one or more courses.

6.1. System model

Given a heterogeneous dissemination model, without loss of generality, we model the network as having two parts:

(i) The CSP's core network, where typically high bandwidths are available. We assume that links in the core network are *provisioned* such that bandwidths in the range of 384 kbps to 1 mbps are available for such an application. We term the edge nodes in the core network as *region nodes*.

(ii) The access network through which the clients connect to the region nodes. Typically bottlenecks occur here.

We consider the CSP as the source, responsible for content dissemination as well as network management. We assume the following: (i) core network topology remains constant (ii) link bandwidths in the network remain static for the duration of the play out. (We relax this assumption in our on going work) (iii) clients attach to region nodes (a maximum of three hops separate a client from a region node).

We consider the following two transcoding node selection criteria given this system model: (i) transcoding possible at all region nodes (SNT-1), as all clients connect to the network through a region node and (ii) transcoding possible only at relay nodes having multiple outgoing links in the core network (SNT-2), as these relay nodes serve two or more region nodes. Even though we have considered only two selection options in this paper, note that other options of choosing combinations of nodes based on client history and CSP's investment constraints can be given as input to the algorithms.

Figure 5 illustrates two topologies that emerge based on clients joining for a course, and the region from which they join.

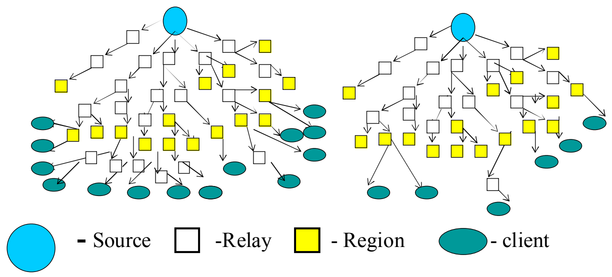


Figure 5: Topologies for two sessions-
Core network remains constant

6.2. Experiments

We derive our topology parameters from a snapshot of the Gnutella peer network [1], having 510 nodes and 14 levels. Note that we are using a single source approximation of the Gnutella peer network.

In AT all relay nodes in the entire network are assumed to have transcoding capability. Thus, AT would provide service to most clients. However, AT is not practical for the following two reasons: (i) it is expensive and redundant to enable all relay nodes with transcoding capability and (ii) CSP may not have control over the access network nodes. We use AT as a benchmark and evaluate our two simple strategies SNT-1 and SNT-2 under various network topologies and client requirements. We consider the following performance parameters:

(i) *number of clients serviced*

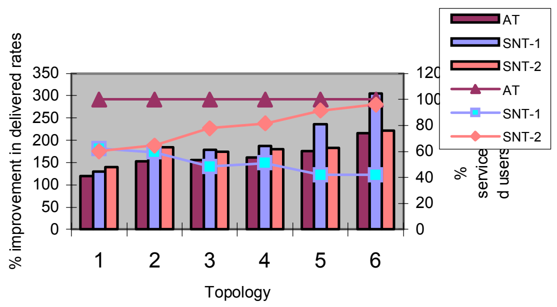
(ii) *average percentage improvement* in delivered rates at clients as compared with their minimum requested rates, and

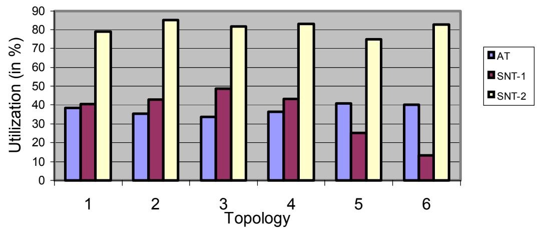
(iii) *utilization of transcoding resources* which is defined as the percentage of deployed transcoders that are actually used.

In our first set of experiments, we generate 6 different Gnutella peer-like topologies having total of 510 nodes and 12-14 levels, starting with 200 nodes in the core and increasing the size of core by 50 nodes up to 450 nodes. Note that the access topologies have 310, 260, 210, 160, 110, and 60 nodes respectively connected to randomly chosen region nodes through access network of up to three levels. Considering that typically the links in the CSP's core network are highly provisioned, we randomly assign link bandwidths in multiples of 64 kbps such that: (i) in the core network link bandwidths are chosen from 384 kbps to 1152 kbps (ii) in the access network, links are assigned bandwidths selected from 128 kbps to 384 kbps. We set all client requirements to 128 kbps of minimum rate and ½ hr of delay tolerance.

In Figure 6(a), we plot using a bar chart, the percentage improvement in rates along Y-axis, and using a line graph, the number of clients serviced, along the secondary Y- axis. We find that SNT-1 and

SNT-2 perform very closely for topology 1 when the core topology is small compared with the access topology. However, when fewer clients join through the access network, having transcoding capability at all region node becomes redundant. In this case, as shown in Figure 6(a), SNT-2 performs as well as AT in delivering enhanced quality to clients while using 63% less transcoders. We find that utilization of SNT-2 is consistently better than AT and SNT-1 as illustrated in Figure 6(b), as more of the deployed transcoders are used by SNT-2.

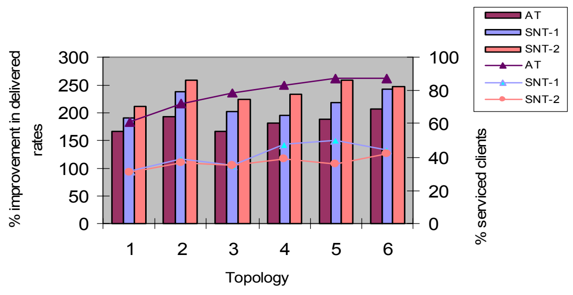


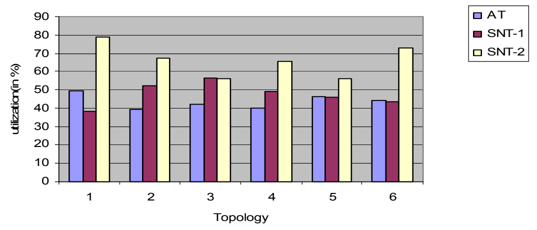


Figures 6(a), (b): Evaluation of AT, SNT-1, SNT-2
Random topologies, same client requirements

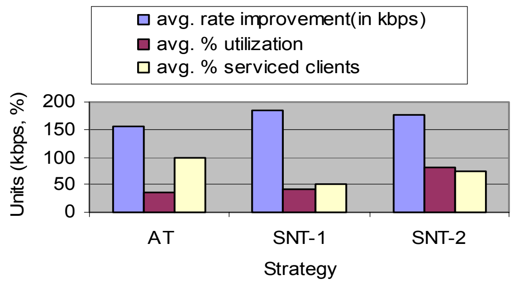
We repeat the experiments with randomly chosen client requirements: minimum rates from 64-256 kbps; delay tolerance values are chosen from 5min.-1 hr. As shown in Figure 7(a), we find that SNT-1 and SNT-2 perform closely; however SNT-2 utilizes the transcoding resources better compared to SNT-1 as seen from Figure 7(b). Note that given the investment constraints of the CSP and client profiles predicted from history, a combination of appropriate relay and region nodes can be selected as input to our algorithm.

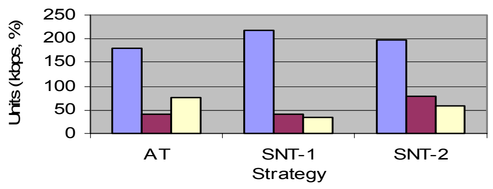
In the second set of experiments, we consider a core topology with 200 nodes and randomly generate access networks having 50, 70, 90, ., 250 nodes. In these experiments, the transcoder placements *remain constant* for SNT-1 and SNT-2. Our objective is to study the impact of size of access network on the performance of SNT-1 and SNT-2 as compared with AT. We first assume that all clients have uniform requirements: minimum rate of 128 kbps and delay tolerance of ½ hr.; we then randomly generate client requirements as in experiments 1.





Figures 7(a), (b): Evaluation of AT, SNT-1, SNT-2
Random topologies, random client requirements





Evaluation of AT, SNT-1, SNT-2
Figures 8(a): Constant core, same client requirements
8(b): Constant core, random client requirements

In Figure 8(a), we find SNT-2 performs better than SNT-1 for both metrics: average utilization and average serviced clients. The average delivered rate in SNT-2 is less as it serves more clients with smaller improvements (which affect the overall average rate delivered).

We repeat the experiments with randomly chosen client requirements: minimum rates are chosen from 64-256 kbps; delay tolerance values are chosen from 5min.-1hr. Figure 8(b) compares the average rate improvement, average % utilization, and average % serviced clients for the three strategies. We again find that SNT-2 edges out SNT-1 in performance.

6.3. Related work

Multimedia dissemination is a well-researched topic with many mechanisms proposed for effective and efficient distribution of multimedia data [4][5][12]. Most of the existing literature treats multimedia applications as soft real time applications where play out of the file begins as soon as the client requests for the data. Typically research has focused on reducing the startup latency --- time lag due to buffering at the client to control jitter during the play out. In contrast, we are focusing on delay-tolerant multimedia applications. While our work may be classified under research dealing with placement of Transcoding enabled proxies (TeC) [10], it differs from existing work in this area, as we deal with delay-tolerant applications.The work in [2] deals with broadcasting multimedia products in the spare bandwidth available in a television channel. Customers can be provided with a set of delivery options corresponding to the time of delivery, which is similar to the notion of delay tolerance. This work focuses on maximizing profits for the e-commerce merchant. While we also deal with revenue maximization for the CSP, in this paper we focus on transcoder placement strategies- placement at relay nodes, bit rate conversions they need to perform- to enhance the delivered rates at clients in a heterogeneous multicast environment.

More recently, the work proposed in [3] shows that in a multicast streaming environment, performing transcoding at intermediate nodes is more efficient than transcoding at the source or clients. This work focuses on finding the lowest cost streaming path from the source to a set of clients, given that some intermediate nodes have transcoding capability. It does not consider the available bandwidth from the source to the clients and the delivered quality at the clients. This work reinforces our claim that placement of transcoding capability at appropriate relay nodes would increase the effectiveness of the resource usage.

7. Conclusions and future work

In this paper we consider delay tolerant multimedia applications and discuss algorithms to provide enhanced rates to clients by exploiting their delay tolerance. Transcoders are required at the relay nodes to provide enhanced rates to clients. However, additional costs of transcoders have to be considered for an appropriate decision. We show that Selected Node Transcoding (SNT) is often the best option for effective and efficient use of transcoding resources. Our greedy heuristic based algorithms provide the rate conversions needed at the transcoding nodes and the delivered stream rates at the clients. We observe that,

* If we denote the basic cost of transcoding facility at the source by C_{base}, cost for ST is given by: $C_{ST}= C_{base}$.

* Cost for implementing anywhere transcoding, $C_{AT} = C_{base} + N*C_U$, where N is the number of relay nodes in the network and C_U is the unit cost for providing transcoding capability at a node.

* In SNT, cost can be substantially less than C_{AT} even with a simple SNT option as demonstrated in this paper. $C_{SNT}= C_{base} + n*C_U$, where $n<N$ and C_U is the unit cost for providing transcoding capability. While cost can be reduced with SNT, delivered rates and number of serviced users may suffer, which reduce the CSP's revenue. Considering the cost-quality trade off, our final aim is to build a decision tool for the CSP for efficient use of resources that maximizes its revenue. Our on going work extends this work focusing on maximizing revenues for CSP.

Acknowledgement: We thank Annanda Th. Rath for his help in implementing the random topology generator.

10. References

[1] An atlas of cyber spaces http://www.cybergeography.org/atlas/more_topology.html

[2] C.C. Aggarwal, M.S. Squillante, J.L. Wolf, P.S. Yu, J. Sethuraman, "Optimizing profits in the broadcast delivery of multimedia products", Fifth International workshop on Multimedia Information Systems, October 1999.

[3] A.Henig and D. Raz, "Efficient Management of Transcoding and Multicasting Multimedia Streams", *Integrated Management (IM) 2005*, Nice, France.

[4] J. Liu, B. Li, "Adaptive Video Multicast over the Internet*", IEEE Multimedia*, January-March 2003.

[5] S. Krithivasan, "Mechanisms for Effective and Efficient Dissemination of Multimedia", *Technical report,* September 2004. URL: www.it.iitb.ac.in/~sarask

[6] S. Krithivasan, S. Iyer, "To Beam or to Stream: Satellitebased vs. Streaming-based Infrastructure for Distance Education*" EdMedia,* June 2004.

[7] S. Krithivasan, S.Iyer, "Enhancing Quality of Service by Exploiting Delay Tolerance in Multimedia Applications", *ACM Multimedia*, Nov. 2005.

[8] S.Roy, M.Covell, J.Ankcorn, S. Wee, and T.Yoshimura. "A system Architecture for Managing Mobile Streaming Media Services*", Mobile Distributed Computing Workshop*, May 2003.

[9] Y. Shang, M. P.J. Fromherz, and T. Hogg, "Complexity of Continuous, 3-SAT-like Constraint Satisfaction Problems", *IJCAI-01Workshop on Stochastic Search Algorithms*, Aug. 2001.

[10] B. Shen, S.J.Lee, "Transcoding-enabled Caching Proxy for Video Delivery in Heterogeneous Network Environments", *Internet and Multimedia Systems and Applications* (*IMSA)*,2002, pp 360-365.

[11] B.Shen, S.Roy, "A very fast video special resolution reduction transcoder", *Proc. International Conf. On acoustics speech and signal processing (ICASSP)*, May 2002.

[12] X. Wang, H. Schulzrinne, "Comparison of Adaptive Internet Multimedia Applications", *IEICE Transaction Communication*, VOL.ES2-B, NO.6, June 1999.