End-to-End 'Data Connectivity' Management for Multimedia Networking

K. Ravindran

City College of CUNY and Graduate Center,
Department of Computer Science,
Convent Avenue at 138th Street,
New York, NY 10031, USA
ravi@cs.ccny.cuny.edu

Abstract. The paper describes a management-oriented model for cost-effective 'data connectivity' provisioning between the end-point entities of networked multimedia applications. The 'connectivity' service provider (SP) may maintain multiple policy-based protocol mechanisms that differ in the bandwidth allocation strategies exercised on transport networks and the extent of QoS guarantees enforced for application-level data flows. The required QoS is prescribed through a service interface, with the SP instantiating one of the policy modules with appropriate parameters to meet the QoS requirements. The model allows dynamic switching from one policy module to another, based on a *cost* associated with bandwidth usage by the network infrastructure for a given QoS offering. The management functions of SP monitor the changes and/or outages in network bandwidth in a dynamic setting, and map them onto connectivity costs incurred by the selected policy mechanism. To accommodate this end-to-end connectivity management, the SP employs an extended form of 'diffserv'-style traffic classification for flow aggregation purposes and 'intserv'-style resource control for bandwidth allocation purposes.

1 Introduction

The provisioning of end-to-end 'data connectivity' may be viewed as a service offered over the underlying network infrastructure. Thereupon, clients may build higher level multimedia-oriented services: such as image downloads, real-time video transport, and mining of time-sensitive data. The connectivity service provider (SP) may set up end-to-end paths between data aggregation points to carry the traffic — say, between New York and London. Individual clients may then exchange high volume information over these data paths for sports, business, and entertainment applications. The SP may possibly lease the bandwidth from infrastructure networks (such as telecom companies) for providing the session-level 'data connectivity' between end-points.

In providing 'data connectivity', the SP is faced with two conflicting goals: reducing the bandwidth costs incurred on the network infrastructure for data flows (to maximize the SP's revenues) and allocating enough bandwidth to meet the

J. Dalmau and G. Hasegawa (Eds.): MMNS 2005, LNCS 3754, pp. 190–203, 2005.

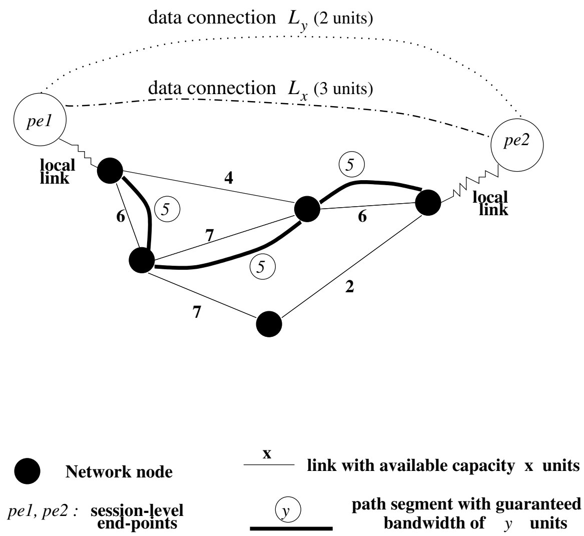

Fig. 1. Bandwidth-controlled connectivity

QOS needs of application sessions (to satisfy the end-user's utility). The SP needs to implement policy mechanisms and management tools that allow balancing these goals. In this paper, we identify a model of connectivity management that allows attaining the revenue and QoS objectives.

Figure 1 illustrates a session-level data path set up between two end-points pe_1 and pe_2. Each segment in the path may be a native communication link between the routers of an IP network or a TCP (or UDP) connection set up between the nodes of an overlay network. Or, the entire path between pe_1 and pe_2 may be a leased line with dedicated bandwidth. Regardless of the network infrastructure, the end-system treats the data path between pe_1 and pe_2 as a single object for the purpose of bandwidth management and admission control.

The SP may employ a control architecture based on *data flows* and *path guarantees* to exercise end-to-end QOS control. It involves:

- Maintaining multiple *diffserv*-type data paths between the end-points with parameterizable QoS differentiation between them;
- Admission control at the end-points with *intserv*-type bandwidth management over the data paths.

The admission control function in an end-system aggregates a large number of data flows with closely-similar QOS needs over a single path. The traffic correlations that exist among such flows allows reaping the statistical multiplexing gains in bandwidth. The path maintenance function in the end-system suitably apportions the available infrastructure bandwidth between the various paths that carry (aggregated) data flows with distinct QoS levels. This bandwidth apportionment allows the SP to enforce per-flow QoS guarantees.

Referring to Figure 1, the available bandwidth between pe_1 and pe_2 is 6 units. Out of this, 5 units are allocated to carry QoS-controlled data flows and the surplus 1 unit is allocated to carry, say, 'best-effort' traffic. The 5 units of bandwidth may in turn be split across two data connections, say, 3 units along L_x and 2 units along L_y to carry high resolution and low resolution video traffic respectively. Here, the SP-level control is about deciding how to estimate the bandwidth of 5 units needed for video traffic and how to split this bandwidth as 3 units and 2 units for L_x and L_y respectively.

The SP may use *policy* functions that prescribe how distinct the flow specs characterizing various data connections are and what cost the per-flow bandwidth usage on a data connection incurs. The SP may also dynamically switch from one policy to another, based on how the costs of bandwidth usage vary as the connection operating point changes (say, due to bandwidth outages in the underlying path). Our model allows installing a repertoire of policy functions at end-points and selecting a suitable policy to make the connectivity provisioning cost-optimal. The paper provides the functional mechanisms to realize the policy switching while sustaining a user-transparent QoS control. These mechanisms are based on our studies on different types multimedia data connections. Overall, our connectivity model can be incorporated into the 'telecommunications management' framework (TMN) that has been standardized for network services [1].

The paper is organized as follows. Section 2 describes a QOS-oriented view of 'data connectivity' and how 'data connections' are managed by the end-system in our model. Section 3 identifies the end-point mechanisms and infrastructure interfaces to support the model. Section 4 compares our approach with existing methods of connectivity control. Section 5 concludes the paper.

2 Our Model of End-to-End Connectivity Management

A session-level connectivity is based on setting up one or more 'data connections' between a pair of peer end-points. The set of links that provide the physical connectivity between end-points constitutes the 'infrastructure', and the available link capacities in a path connecting the end-points constitute the 'resource'. An admission control protocol exercises the bandwidth to sustain a certain QoS of data transfer over the connections.

In this section, we provide a management-oriented view of the end-to-end mechanisms that exercise bandwidth allocation control.

2.1 Management-Oriented View of Connectivity Protocols

A 'data connection' is characterized by QoS attributes: such as the sustainable rate of data flow, maximum allowed data loss, and end-to-end delay jitter on data [2,3]. The client application prescribes this QoS to the 'connectivity service' provider (SP) when requesting the setup of a 'data connection'. The internal functions of a SP's infrastructure that realize end-to-end data connectivity, such as packet scheduling strategies, are however hidden from the application.

A connectivity protocol $\mathcal{P}$ encapsulates the functionality to manage the end-to-end admission of data traffic and the required provisioning of infrastructure bandwidth. The QoS parameters q instantiate this functionality at run-time to control the extent of bandwidth allocation. The bandwidth expended b may be represented as: $b = \mathcal{R}(x_{\mathcal{P}})$, where $x_{\mathcal{P}}$ is the protocol-internal state that reflects the current operating point of the connection (such as queue sizes, window credits, etc). For instance, $\mathcal{R}$ may depict a mathematical formula for 'link utilization' achieved by a 'window-based data transfer' protocol, expressed in terms of the window sizes and link error rate. Such a functional representation allows a management module to maintain a handle on the bandwidth allocation exercised by $\mathcal{P}$ on 'data connections'.

2.2 Connectivity-Level Objects for Management Control

The management control is exercised on two types of session-level objects: 'data flow' and 'data connection'. A 'data flow' is a sequence of packets transported from the source to receiver entities, subject to a certain end-to-end QOS. A 'data connection' is set up over the transport path between source and receiver entities, with a prescribed amount of bandwidth allocation to carry a group of data flows with a closely-similar QOS characteristics. See Figure 2. A 'data connection' is the object granularity for bandwidth allocation purposes, whereas a 'data flow' is the object granularity for end-to-end admission control.

Given a bandwidth apportionment $b(r)$ for a data flow r, the amount of bandwidth usage $\sum_{\forall r} b(r)$ incurred by a 'data connection' C can be transcribed into a cost of transporting various data flows $\{r\}$ over C. This cost may depend

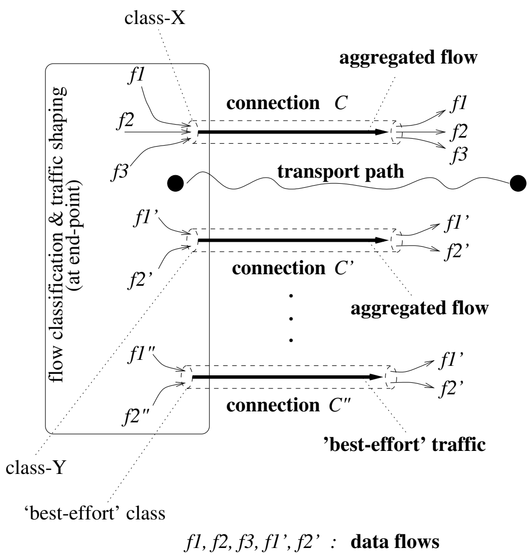

Fig. 2. 'Data connection' versus 'data flows'

on, say, the infrastructure-level tariffs incurred for bandwidth allocation[1]. The SP attempts to multiplex many data flows on a single connection to reap the gains arising from a statistical sharing of the bandwidth — and hence reduce the costs. This revenue-oriented incentive forms the basis for a dynamic control of connectivity mechanisms employed by the SP.

2.3 Application-Level Flow Specs

The connectivity protocol embodies a policy function $\mathcal{F}$ to map a data flow r to the bandwidth needs b at network elements in a data path. In one form, r may be given by a peak rate p, average rate A, loss tolerance limit Δ (specified as a fraction of the average rate in the range $[0.0, 1.0]$), delay tolerance limit $\mathcal{D}$, and auto-correlation parameter ζ of data traffic. Note that $\zeta \in (0.0, 1.0)$, with $\zeta \rightarrow 0.0^+$ indicating a totally random flow and $\zeta \rightarrow 1.0^-$ indicating a high degree of statistical dependence of the current peak rate on past peak rates. $\mathcal{F}$ maps a data flow r to bandwidth needs b such that $\mathcal{F}(r') > \mathcal{F}(r'')$ for $r' > r''$. See [4] for guidelines to prescribe the '>' relation on flow types f.

Consider a data flow r of type $f = (A, p, \zeta, \Delta, \mathcal{D})$ over a network element E, where $A < p$ and $\Delta > 0$. A policy function[2] $\mathcal{F}$ may employ *optimistic* bandwidth allocation on E by assuming that the peak rate of flow does not persist long enough to backlog packets at the input queue of E to a level where more than a fraction Δ of the packets will miss their deadlines prescribed by $\mathcal{D}$. Such an allocation will have: $[A - \Delta] < \mathcal{F}(r) < p$, with the actual allocation determined by $\mathcal{D}$, the duration of p relative to A (i.e, burstiness), ζ, and input queue length of E. If $\mathcal{F}$ and $\mathcal{F}'$ depict optimistic policies such that $\mathcal{F}(r) > \mathcal{F}'(r)$ for some $r \in$ FLOW_SPECS, then $\mathcal{F}(r') > \mathcal{F}'(r')|_{\forall r' \in \text{FLOW\_SPECS}}$.

Note that the flow type f may be viewed as a 'traffic class' in the DiffServ architecture. A connection $C(f)$ is then a 'DiffServ' path to carry data flows of type f. The apportionment of available bandwidth B on a network path across the various connections sharing this path corresponds to a 'proportional differentiation' in the scheduling of packets of these traffic classes [6].

2.4 Policy-Based Estimation of Bandwidth

$\mathcal{F}$ encapsulates a resource allocation policy realized at the end-points. Typically, an allocation may be somewhat less than supporting the peak rate p in a sustained manner, but more than the average rate A, with the constraint being that the packet loss over the observation interval T_{obs} is less than Δ. An example of allocation policy is to reserve 10% additional bandwidth relative to that necessary to sustain the average rate A. Typically, the scheduler should visit the

[1] The SP may possibly lease fiber-optic link-level connectivity between end-points from telecom companies (such as AT&T), and then control bandwidth allocation on this leased link to support session-level 'data connectivity' for customers.

[2] The (p, A, Δ, ζ) tuples may be viewed as prescribing distinct 'virtual link classes' (see [5] in this context). The admission controller then maps an application-generated data flow to one of these 'virtual links'.

packet queue of C for a portion $(\frac{b}{\text{CAP}(E)} - \Delta)$ of T_{obs}, where b is the effective bandwidth necessary for a no-loss transfer of packets over the link.

The resource encapsulation embodied in $\mathcal{F}$ allows different policy functions to be installed at the management interface points of a 'data connection'. Our focus here is not on the accuracy in estimating the bandwidth needs itself, but is on the signaling support for a reasonable estimate from the traffic-oriented QoS parameters. It may be noted that the IntServ-style bandwidth allocation embodied in our model is exercised only at the end-system[3].

2.5 Revenue Incentives of Flow Aggregation

One or more data flows may be multiplexed over a single connection C. The multiplexing may reap bandwidth gains due to statistical sharing of the bandwidth allocated for C across these flows. Typically, the SP may multiplex flows with similar QoS characteristics over C so that all packets of these flows get the same level of scheduling. For instance, multiple MPEG-2 video streams may require the same level of loss/delay tolerance [9], which makes the video packets schedulable with a single-level priority scheme. Such an aggregation allows incorporating the gains arising from a statistical sharing of connection bandwidth in the SP's revenue-oriented decisions.

In 'one-at-a-time' floor-controlled voice conferencing, the voice data bursts from various speakers may be spaced in time. This correlation in turn allows keeping the bandwidth allocation on C to just sustain a 64 *kbps* data rate for all the voice streams combined, that otherwise would be higher if allocations are done separately for each voice stream.

Consider a system of sensors that collect data pertaining to a common external phenomenon (e.g., multiple radars observing a plane flying over a terrain). The data collected by various sensors may exhibit a high degree of correlation in the traffic behaviors, since these data pertain to the same physical phenomenon. A traffic shaper at end-points can spread out the peak rates of sensor data in a controlled manner to achieve a steady but lower bandwidth consumption.

Thus, an aggregation of closely-similar data flows over C offers the potential for statistical multiplexing gains. Higher the number of flows multiplexed over C, more are the bandwidth gains. The gains however accrue at the expense of a certain amount of packet loss.

As can be seen, a 'data connection' offers the right granularity to enforce bandwidth allocation policies by the SP. A management perspective of the end-system mechanisms to control such 'connection' objects is described next.

3 End-System Protocol Mechanisms

The infrastructure mechanisms are built around 'packet scheduling' over data connections, weighted by their bandwidth allocations. The packet scheduler is

[3] The IntServ-type and DiffServ-type of functional elements in our end-system model are inspired by, but are different (both in context and scope) from, the IntServ and DiffServ architectures proposed by IETF for use in the core network elements [7,8].

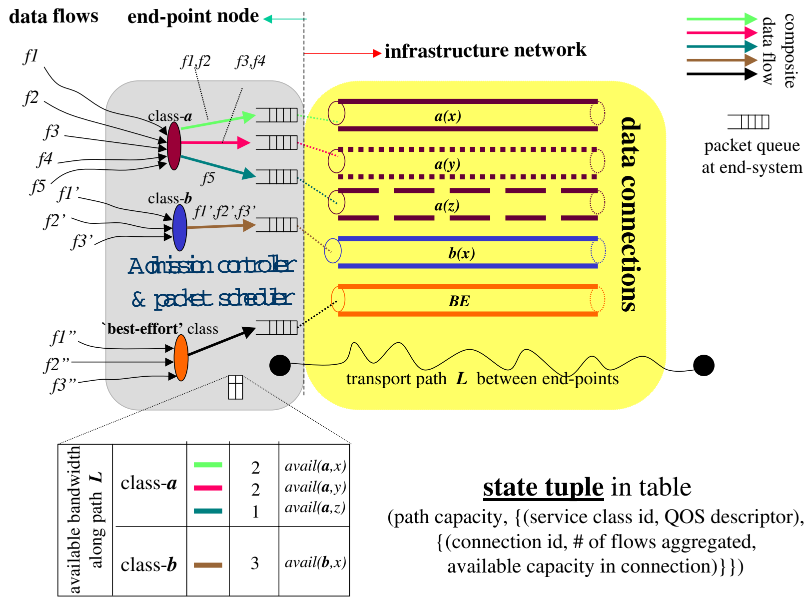

Fig. 3. State maintained at end-point nodes

a canonical end-point element that interfaces between the connectivity protocol $\mathcal{P}$ and the network infrastructure. There is however no per-flow state tracking at the infrastructure level[4].

3.1 State Information at End-Points

Aggregating multiple flows over a single connection C reduces the scheduling overhead, relative to setting up a separate connection for each data flow. Figure 3 illustrates the state information maintained at end-points to support flow aggregation. The key pieces of state information include the QoS specs that classifies the component flows, number of flows multiplexed, policy function to map QoS specs to bandwidth needs, and available bandwidth on a connection. Since this information is maintained at connection-level, the amount of per-flow state is reduced by $\mathcal{O}(n)$, where n is the number of flows aggregated over C. The only per-flow control activity incurred at the admission controller when flows are admitted or removed is to adjust the number of flows n and re-estimate the bandwidth needs using policy functions[5].

[4] We assume a FIFO based intra-connection scheduling across the component flows. This ensures the scalability of end-point mechanisms by avoiding the need for per-flow state-tracking.

[5] Flow aggregation in our model is primarily for revenue-oriented bandwidth management purposes (besides achieving scalability of protocol-level implementations of admission control). The per-flow guarantees experienced by end-to-end peer-entities is a by-product of this paradigm shift in bandwidth management.

To enable the aggregation of data flows, the session-level manager may assign a unique label $l(C)$ to bind the component flows together, whereupon the admission controller can multiplex them over C. Here, $l(C)$ is a session-level index to the grouping of data flows that are carried over C. In the example of sensor system, $l(C)$ may be the id referring to the external phenomenon from which the sensor data are collected. The session-level labeling of connections can be part of, say, a MPLS-based routing [10] over the path set up through the network.

3.2 Determination of Bandwidth Savings

A ’data connection’ shared across many closely-similar flows entails a lower per-flow bandwidth allocation that is quantifiable, in comparison to a case of sharing the underlying network path across many disparate flows. The bandwidth allocation over a shared ’data connection’ satisfies *weak additivity*, indicated as:

$$\mathcal{F}(r_i)|_{i=1,2,\cdots,n} < \mathcal{F}(r_1 \oplus r_2 \oplus \cdots \oplus r_n) \leq \mathcal{F}(r_1) + \mathcal{F}(r_2) + \cdots + \mathcal{F}(r_n),$$

where $\mathcal{F}(r_i) > A_i$ and ’$\oplus$’ is the aggregation operator. This relation captures the possible savings due to sharing of connection-level bandwidth across various flows, with the actual gains determined by the cross-correlation parameter associated with these flows. Such an end-point admission control procedure is illustrated below:

admit_flow$(r_{i+1}, q, C)|_{i=1,2,\cdots}$ /* q: flow descriptor for connection C */
  additional bandwidth $X := \mathcal{F}(r_1 \oplus \cdots \oplus r_i \oplus r_{i+1}) - \mathcal{F}(r_1 \oplus \cdots \oplus r_i)$;
  if ($X <$ availbw(C)) /* enough bandwidth is available */
    admit new flow r_{i+1};
    availbw(C) := availbw(C) - X;
  else
    reject new flow r_{i+1}.

When there is no connection-level sharing, the inability to map the traffic correlation onto the packet scheduling exercised on various data flows forces the end-system to determine the bandwidth needs independently for each of the flows. So, the total allocation is $\sum_{i=1}^{n} \mathcal{F}(r_i)$. This in turn precludes bandwidth savings that may otherwise be feasible due to a shared allocation driven by traffic cross-correlation across the data flows.

In a general form, the per-flow bandwidth allocation may be given as:

$$\mathcal{R}_{bw}(n) = \frac{\mathcal{F}(r_1 \oplus r_2 \oplus \cdots \oplus r_n)}{n}.$$

The monotonicity condition that depicts the bandwidth sharing is: $\mathcal{R}_{bw}(n) < \mathcal{R}_{bw}(n')$ for $n > n'$. Figure 4 illustrates how a policy function $\mathcal{F}$ may

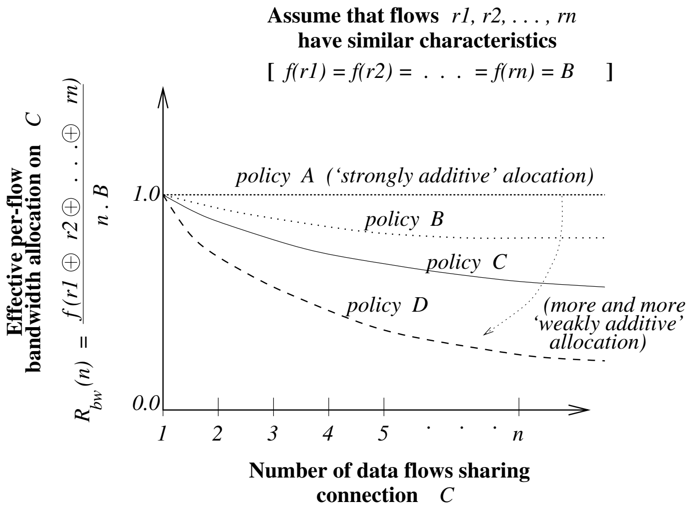

Fig. 4. Bandwidth allocation policies for shared connections (an empirical view)

capture these gains, so that it can be plugged in by the SP at appropriate control points[6].

3.3 Packet Delay and Loss Checks

Packet delay checks are made against flow-specific delay tolerances. However, a 'connection' is the object granularity seen at the scheduler level. The scheduler may use the connection id (cid) carried in packets to index them into appropriate queues and exercise packet scheduling therefrom. Since only flows with similar characteristics are multiplexed over a connection, delay checks at connection-level can provide information about packets meeting flow-specific delay tolerance parameter $\mathcal{D}$. The 'delay comparison' relation for a packet p is:

$$p.\text{timestamp} + \mathcal{D}(p.\text{cid}) \;>\; \text{current\_time} + T_{tx},$$

which qualifies p as meeting the deadline at receiver. Note that an excessively delayed packet is deemed as a lost packet for end-to-end control purposes. The tolerance parameters $\mathcal{D}$ and Δ are passed on to the admission controller through a signaling mechanism for use by the scheduler.

[6] It is not the mechanism of 'statistical multiplexing' that we focus in the paper. Rather, it is how we can quantitatively represent the policies that reap 'statistical multiplexing' gains, so that these gains can be factored into the flow admission decisions by the SP. In this light, how effective a statistical multiplexing scheme is and how good a quantitative representation of the multiplexing gains is are orthogonal aspects. The former pertains to a traffic engineering based protocol design, whereas the latter deals with how to incorporate the gains in a macroscopic policy function for use in revenue-based decisions.

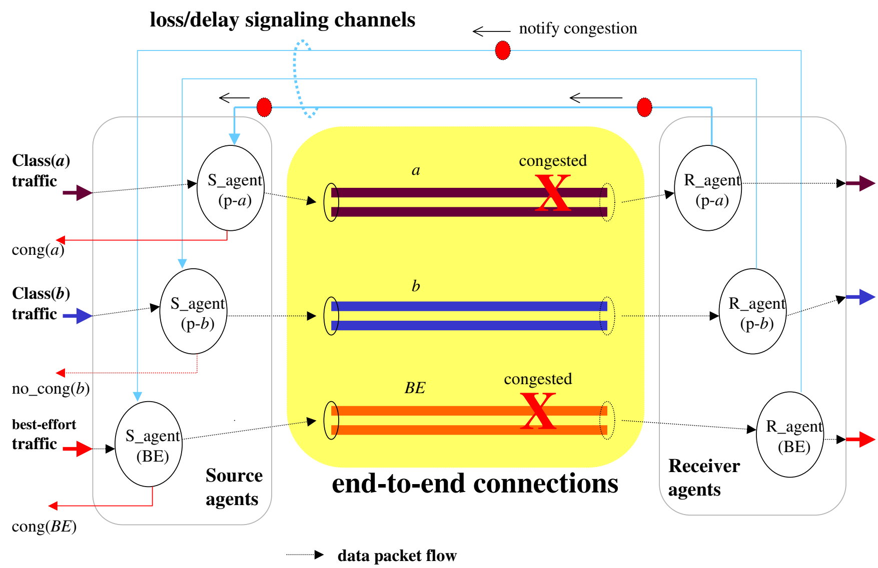

Fig. 5. Agent-based connection monitoring

Figure 5 shows an agent-based implementation of the monitor for packet loss and delays. Here, congestion on a data connection C may arise due to a possible inability of the admission controller to determine the allowed levels of flow multiplexing over C prior to actually admitting the flows. In our study, the signaling of packet loss from the agents at receiver end to the agents at source end is carried out using IETF RTCP. The signaling delay on a 4-hop network is measured about 30 $msec$.

3.4 Optimal Level of Multiplexing

The multiplexing of data flows over a connection C may affect client-level QoS due to excessive path sharing among data flows and sustained higher rates in many of them. Also, the intra-connection multiplexing and de-multiplexing overhead on packets — which is another form of cost (besides bandwidth cost) — increases with the number of flows multiplexed on C. Figure 6 shows this relationship in an empirical form. The work in [11] has shown that the queuing delay of packets is a monotonically increasing function of the number of flows n that feed packets into the queue. Thus, beyond a certain level of bandwidth sharing (say, for $n > n''$), the end-to-end delay of packets belonging to various flows may increase to a level where the client-prescribed loss tolerance limit Δ is not met. This packet loss behavior is depicted as:

$$\mathcal{R}_{loss}(n) = \frac{\sum_{i=1}^{n} p_i - \mathcal{F}(r_1 \oplus r_2 \oplus \cdots \oplus r_n)}{n},$$

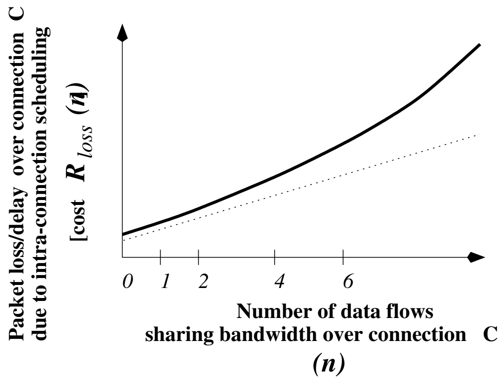

Fig. 6. Intra-connection scheduling costs (an empirical view)

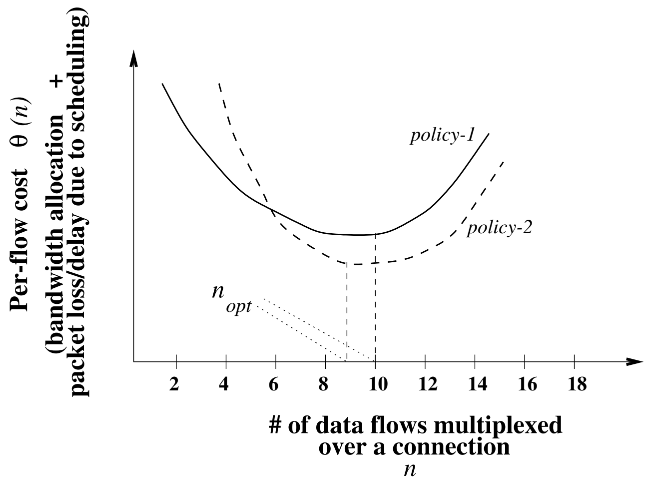

Fig. 7. Combined intra-connection bandwidth and scheduling costs

where p_i is the peak rate of flow r_i. The monotonicity condition is: $\mathcal{R}_{loss}(n) > \mathcal{R}_{loss}(n')$ for $n > n'$.

In general, the per-flow bandwidth cost $\mathcal{R}_{bw}(n)$ on a connection can be reduced by increasing the number of flows sharing this path. The lower bandwidth usage may however be counteracted by increased packet loss $\mathcal{R}_{loss}(n)$ arising from scheduling delays. Accordingly, the number of flows admitted in C should not exceed a threshold n_{opt} that may cause connection failures due to excessive packet loss. See Figure 7. To enable determination of this optimal point at run-time, the SP prescribes a cost function of the form:

$$\Theta(n) = a.\mathcal{R}_{bw}(n) + b.\mathcal{R}_{loss}(n)$$

for use by the admission controller, where a and b are normalization constants. Since there is no closed-form analytical relation between $\Theta(n)$ and n, the optimal value n_{opt} needs to be determined dynamically by measurements of packet loss experienced over C at run-time. That $\mathcal{R}_{BW}(n)$ and $\mathcal{R}_{loss}(n)$ exhibit monotonicity properties ensures that the $\Theta(n)$-versus-n relation has a single global minimum, and hence allows determining n_{opt} by an iterative search procedure.

With multiple policy functions available for the SP, empirically relating them in terms of cost allows the SP to dynamically switch from one policy to another. If $\mathcal{F}$ and $\mathcal{F}'$ depict policies such that $\mathcal{F}(r) > \mathcal{F}'(r)$ for some $r \in$ flow_specs, we then have: $n_{opt}(\mathcal{F}) > n_{opt}(\mathcal{F}')$. The SP thus determines the optimal assignment of groups of data flows to a set of distinct connections[7].

The cost analysis required for connectivity control may be based on separate empirical studies of various policy functions.

4 Related Works

There have been works that attempt to get the advantages of "IntServ" world, namely, flexibility and fair QoS support and that of "DiffServ" world, namely, robustness and scalability. We compare these works with our approach, with an emphasis on SP-level revenue incentives.

Techniques for flow aggregation and avoidance of per-flow tracking have been studied elsewhere: such as the 'dynamic packet state' based packet classification and scheduling by core routers [14] and the 'link-based fair aggregation' for class-based fair queue scheduling at the ingress and egress routers and for intra-class FIFO scheduling at the core routers [15]. Likewise, the admission control architecture in [16] allows the end-points probe the network for bandwidth availability and admit a group of flows only when there is no congestion in the network. Architecturally, these existing techniques for QoS control are based on two session-level objects: 'data flows' and 'bandwidth guaranteed data paths'.

In contrast, our model stipulates another object, namely, 'data connection', to embody the grouping of one or more closely-similar data flows. This in turn allows incorporating statistical multiplexing gains as part of a cost assignment policy to application-level flows. Referring to Figure 1, the 'data connections' L_x of 3 units bandwidth and L_y of 2 units bandwidth simply do not exist in the current models. Instead, only a single end-to-end path of 5 units bandwidth is visible to the session-level controller for multiplexing the various data flows. In this light, the 'data connection' objects in our model offer a better means of quantifying and estimating the bandwidth gains arising from statistical multiplexing of data flows to enable revenue-driven decision-making by the SP's.

In a larger sense, our connectivity model may provide a management dimension to the existing control architectures for end-point flow admissions.

[7] Methods to quantify network resource allocations (such as those described in [12,13]) can be incorporated in policy functions.

5 Conclusions

The paper described a model of session-level connectivity provisioning for use by multimedia networked applications. The model is based on creating a variety of diffserv-type of 'data connections' with QoS differentiation and apportioning the available bandwidth across these connections using intserv-type of end-point admission control. The model employs a policy-driven control of infrastructure bandwidth allocations, for cost-effective provisioning of data connectivity.

Our model allows dynamic switching from one policy function to another, based on a notion of cost associated with the infrastructure bandwidth usage, for a given level of QoS support. The strategy is to reduce the per-flow cost incurred by multiplexing many closely-similar data flows on a single connection. The multiplexing brings two benefits to the SP, without compromising the QoS needs of applications. First, it reduces the per-flow resource allocation due to the gains accrued from a statistical sharing of connection resources. Second, it amortizes the connection-level overhead across many flows. The level of cost reduction, and hence revenue accrual, can be controlled by the SP using a range of policy functions that take into account the QoS attributes of data flows.

Our model accommodates the above strategy through a management-oriented interface that allows the SP to maintain a repertoire of policy functions and choose one therefrom for providing an appropriate level of 'data connectivity' to the client applications. The paper described the functional mechanisms to monitor the end-to-end QOS and adjust the connection operating points to maximize the SP's revenue without compromising the user-level QoS needs.

Our study shows that the connectivity management model can be employed for QoS-sensitive multimedia networks in a scalable and flexible manner.

Acknowledgement

The author acknowledges **Dr. Xiliang Liu** for the discussions on how bandwidth estimation can be incorporated into a service-level management framework for 'data connections'.

References

1. M. Subramanian. Telecommunications Management Network. Chapter 11, Network Management: Principles and Practice, Addison-Wesley Publ. Co. (2000).
2. S. Keshav. Scheduling. Chapter 9, An Engineering Approach to Computer Networking, Addison-Wesley Publ. Co., (1996), 209-260.
3. A. S. Tanenbaum. Congestion Control, Quality of Service, Performance Issues. Chapters 5.3, 5.4, and 6.6, Computer Networks, Prentice-Hall Publ. Co., 4th ed. (2003).
4. J. Wroclawski. Specification of the Controlled-load Network Element Service. Internet RFC 2211 (1997).
5. S. Floyd and V. Jacobson. Link-sharing and Resource Management Models for Packet Networks. IEEE/ACM Transactions on Networking, vol.3, no.4 (1995).

6. C. Dovrolis and P. Ramanathan. Proportional Differentiated Services, Part-II: Loss Rate Differentiation and Packet Dropping. Intl. Workshop on Quality of Service, IWQoS'00, Pittsburgh, USA (2000).
7. S. Berson and S. Vincent. Aggregation Internet Integrated Services State. Internet RFC (1997).
8. K. Nichols, V, Jacobsen, and L. Zhang. Two-bit Differentiated Services Architecture for Internet. Internet RFC 2638 (1999).
9. L. Boroczky, A. Y. Ngai, and E. C. Westermann. Statistical Multiplexing Using MPEG-2 Video Encoders. IBM Technical Journal, vol.43, no.4 (1999).
10. U. Black. MPLS and Label Switching Networks. Prentice-Hall Publ. Co., 2nd ed. (2002).
11. R. Guerin and A. Orda. QoS Routing in Networks with Inaccurate Information: Theory and Algorithms. IEEE/ACM Transactions on Networking, vol.7, no.3 (1999).
12. H. M. Mason and H. R. Varian. Pricing Congestible Network Resources. IEEE Journal on Selected Areas in Communications, vol.13, no.7, (1995), 1141-1149.
13. J. Vicente, H. Cartmill, G. Maxson, S. Siegel, R. Fenger. Managing Enhanced Network Services: a Pragmatic View of Policy-Based Management. Intel Tech. Journal, Q1, (2000).
14. I. Stoica and H. Zhang. Providing Guaranteed Services Without Per-flow Management. Proc. ACM SIGCOMM'99, Cambridge, USA, (1999).
15. Y. Jiang. Link-based Fair Aggregation: a Simple Approach to Scalable Support of Per-Flow Service Guarantees. Tech. Report, Norwegian Inst. of Technology, Norway (2004).
16. L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang. End-point Admission Control: Architectural Issues and Performance. Proc. ACM SIGCOMM'00, Stockholm, Sweden, (2000).