Internet Engineering Task Force
Internet Draft
Expires: September 1999

Hungkei (Keith) Chow
Alberto Leon-Garcia
Network Arch. Lab,
University of Toronto
March 1999

A Feedback Control Extension to Differentiated Services

<draft-chow-diffserv-fbctrl-00.txt, .ps, .pdf>

Status of Memo

This document is an Internet-Draft and is NOT offered in accordance with Section 10 of RFC2026, and the author does not provide the IETF with any rights other than to publish as an Internet-Draft

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This draft presents a Feedback Control extension to Differentiated Services. Differentiated Services have been designed for scalability through handling aggregates of traffic instead of individual flows as in the Integrated Services. However, it has been observed that the DS mechanism in some situations can hardly achieve the desired quality of service and may result in unfair conditions. To remedy these problems, this draft describes a general feedback control paradigm that enables a network provider to impose a control mechanism upon their DS domain. As an instance of the general framework, a feedback control mechanism is proposed. Our simulation analysis demonstrates that the overall feedback controlled DS can offer a better resource utilisation and a fair resource sharing. Such control mechanism can also help enforce the desired service assurances. This document is intended to stimulate discussion in this direction. Further work is required to carefully define a set of primitive requirements that enables interoperability.

The pdf and ps version of this document are available at:
http://www.comm.utoronto.ca/~keith/ietf-id/draft-chow-diffserv-fbctrl-00.pdf, .ps
and recommended for the figures it contains.

1 Introduction

Differentiated services provides different level of network services by employing a set of well-defined building blocks. The mechanism is that a small label (the diff-serv code-point or DSCP) in the IPv4 TOS octet or IPv6 Traffic Class octet is used to determine that a packet is to receive a particular forwarding treatment (per-hop behaviour or PHB) at each network node. At the diff-serv boundary, routers enforce the SLAs by including functionality such as traffic conditioning, monitoring and packet classification, in addition to providing the PHB requirements. Detailed description of diff-serv is given in its architecture [DSARCH], framework [DSFMWK], DSCP specification [DSHEAD] and boundary requirement [DSBOUND] documents.

A salient feature of diff-serv is its scalability. It is achieved by handling aggregated traffic using one or a small number of PHBs within the core network rather than on a per-flow basis, thereby simplifying the processing and storage associated with packet classification and signalling. However, our analysis [THESIS] and reports by other researchers [ASIBN, ASKLT] have shown that diff-serv may result in an unfair and inefficient resources sharing. To remedy these drawbacks, we introduce a feedback control mechanism into diff-serv, namely feedback controlled diff-serv (or FC-DS).

In this draft, we discuss the general paradigm of FC-DS. Section 4 describes a possible instance of the framework and presents some performance results. Section 5 discusses other practical issues related to the proposed framework. We hope that this draft will stimulate discussion within the working group.

2 Motivation

Recent research results [ASIB, ASKLT, ASBW] and our analysis [THESIS] have indicated that under various situations, existing diff-serv mechanisms may have problems of unfairness and inefficient resource utilisation, thereby failing to achieve the desired QoS. Table 1 summarises the potential problems under different conditions.

Conditions	Outcomes	Problems
Excess bw available	▪ Aggressive and/or non-adaptive flows take most of excess bw	▪ Unfair bw share
Insufficient bw	▪ High profile flows will be hit first ▪ Aggressive and/or non-adaptive flows have advantage	▪ Unfair service degradation
Flows with different Round-trip-time	▪ Flows with short round-trip-time have advantage	▪ Some flows cannot achieve assured rates
Flows with different requested profiles	▪ Low profile flows have advantage	▪ Some flows cannot achieve assured rates
Congestion	▪ Sustained when flows are not co-operative ▪ Packets only dropped at congested link	▪ Inefficient use of bw ▪ Larger delay & jitter ▪ More buffer space required

Table 1 : Summary of problems with diff-serv mechanism

Although some forms of call admission control (CAC) mechanisms may help alleviate the problems, we argue that CAC is only a necessary but insufficient requirement. Since the problems are associated with the dynamics of the network load and capacity, it has been shown earlier in the literature that static solutions, such as allocating more buffers, providing faster links or tightening the CAC policy, does not solve the problem.

Generally, there are at least three major causes for the problem:

(1) No isolation of flow inside the core of the network: When flows enter the core of the diff-serv network, they are naturally aggregated and forwarded using one or a few number of PHBs according to their DSCP. In order words, flows are aggregated into one or a few number of shared buffers, each of which is allocated a certain amount of forwarding resources in terms of scheduling or dropping priority. Since flows are indistinguishable (or intended not to be distinguished) within a shared buffer, aggressive flows may deprive other flows of any available resource, thereby resulting in an unfair resource sharing.

(2) No dynamic control at the diff-serv boundary: Once a flow is allowed to enter a DS domain, it is usually policed or conditioned at the ingress node according to its TCA. However, the conditioning function is done in a static manner such that it does not respond to the network dynamics.

(3) Reliance only on transport protocol to react: with presence of non-adaptive flows (e.g., UDP flows), TCP flows generally receive poorer service than UDP flows. This is because TCP sources back off when their packets are dropped, whereas the UDP sources do not react to dropping of their packets. Although RTP/UDP may provide a certain degree of adaptivity, its granularity may not be suitable for network control purposes. Moreover, even for the case of all adaptive flows, recent work [FENG] has indicated that some modifications to TCP are required in order to achieve the desirable service differentiation.

To remedy these problems, we propose a dynamic control mechanism in which the boundary routers periodically obtain information from the core of the network and use this information to update their traffic conditioners. Since a more precise control on the incoming traffic can be achieved at the ingress node, a better resource sharing may be possible at the core of the network. By incorporating this dynamic control mechanism, network providers not only can handle traffic congestion more effectively, but they can also manage their traffic and resources more efficiently.

3 Feedback Controlled Diff-Serv (FC-DS)

It is commonly believed that different network vendors may prefer to deploy their proprietary control mechanisms according to their policy requirements. Our proposed control framework, therefore, should be generic and flexible enough for this purpose. Moreover, it is desirable that it is backward compatible with existing diff-serv mechanism for enabling interoperability.

In considering these requirements, we define a general FC-DS in a way that a variety of control mechanisms can be derived from it. The concept of FC-DS is that the boundary routers periodically probe the core of the network to obtain the current state information. This network information is used by the ingress or boundary routers to update their traffic conditioners such that a more precise control on the incoming traffic can be achieved.

The following sections describe the extensions of the architectural model and framework for constructing the FC-DS. They should be read along with [DSARCH, DSFMWK, DSBOUND].

3.1 Architectural Model

The FC-DS architecture is built based on the DS architecture. It is generally a superset of the requirements and functionality defined in [DSARCH]. In this Section, we define the additional functions required to construct a feedback control mechanism.

3.1.1 FC-DS Domain

A FC-DS domain is a DS domain enhanced with a feedback control mechanism. It is possible that the control mechanism spans across multiple DS domains or within only one domain. In this Section, we consider only the intra-domain control mechanism while a brief discussion on inter-domain control is given in Section 5.2.

3.1.2 FC-DS Ingress node

An ingress node generally performs traffic conditioning functions to ensure that the traffic entering a DS domain conforms to the rules specified in the TCA, in accordance with the domain's service provisioning policy. Since TCA is usually a static agreement, unless re-negotiation is allowed, the traffic profile derived from a TCA is fixed once a flow is accepted. In FC-DS, we propose to make this TC functions adapt to the state of the DS domain. The general rules for the adaptive TC (ATC) are:

(1) Under a normal situation, ATC performs the same TC functions as in the conventional DS;
(2) When excess resources are available inside its domain, ATC should modify its policing function such that traffic flow will have a fair share of the excess resource pool;
(3) When congestion occurs, ATC should ensure that each traffic flow will experience a fair service degradation. This can be achieved by tightening the traffic profile of individual flows in a fair manner; and
(4) All ATC functions should follow the dynamics of the DS domain under control. Therefore, an ingress node is required to have the capability of consolidating reports and then performing the appropriate ATC functions.

Section 3.2.4 further discusses the components of an ATC.

Besides ATC, an ingress node in FC-DS is also responsible for generating probes. A probe is a control packet that is used to collect network information from the DS domain under control. Depending on the control mechanism, the ingress node may send probe packets to its connected interior node(s) on a per-flow or per-boundary node basis.

3.1.3 FC-DS Interior node

In addition to the basic packet forwarding function, an interior node is extended to include a load monitoring function. Upon receiving a probe/report packet, it updates the information carried in the probe/report with its current loading information and then forwards the control packet to other connected node(s).

3.1.4 FC-DS Egress node

For the case of intra-domain control, a FC-DS egress node is where the probe packets are terminated. The egress node is responsible to compose and return a report packet to the ingress node or other boundary node(s), with reference to the received probe packet(s).

Depending on the details of the TCA between two domains, egress nodes may perform traffic conditioning functions on traffic forwarded to the peer domain. For these cases, ATC functions may also be included in FC-DS egress nodes.

It is worth mentioning that the report generation mechanism is not only useful in the context of traffic control, but it also provides a hook for other purposes, such as receiver control [RCVCTRL] and QoS monitoring [NTIMP].

3.2 FC-DS Framework

Having described the extensions of the architectural model, this section details the configurations of the key control mechanisms.

3.2.1 Probe Generation

Generally, the probe generation mechanism is determined by two parameters: probing period and granularity. Probing period refers to how frequently a probe packet is generated or the *temporal resolution* of the control mechanism. Typically, it can be specified in term of a time interval or packet count. The choice of a probing period is related to the dynamics of the DS domain under control as well as the variation of the incoming traffic. To obtain a higher control precision, the ingress node may choose a shorter probing period, i.e. generate probe packets more frequently. However, this probing frequency should be balanced with the amount of processing power required at the network nodes.

Probing granularity, however, refers to the resolution of the control mechanism in *spatial* domain. The following lists some possible examples:

- *Per-aggregated-flow or per-microflow basis*, in which one probe is generated per contracted incoming flow. It implies a flow based control mechanism, which can generally give the finest grain control precision.
- *Per-BA basis*, in which one probe is generated per behavioural aggregate (PHB). If an ingress node has access to more than one PHBs, multiple probe packets will be generated in each probing interval.
- *Per-egress-node or per-boundary-node basis*, in which one probe is generated per boundary node. Notice that the notion of ingress-egress-pair is defined only when there is a flow. Therefore, this probing scheme can be regarded as a topology based control mechanism in which each boundary (ingress) node keeps the statistics of all possible paths having other boundary nodes as egress points.
- A combination of above. Depending on their control algorithm and, particularly, their required control precision, network providers may choose to have a variant or a combination of the above mentioned schemes.

In general, in choosing a probing granularity, one may consider (1) the required control precision, (2) the processing capability of the routers, and (3) the amount of tolerable control overhead.

3.2.2 Probe/Report creation and handling

Since probe/report (control) packets are sent on the same link, an interior/egress node needs a mechanism to distinguish the control packets from other data packets. Several possible alternatives exist for constructing a control packet such that it can be easily identified. They include:

1. Creating a new packet with a special DSCP, in which control information is carried in the data area of the packet.
2. Extending the IP header of a selected data packet using IP header extensions, in which a special extension is defined for carrying the control information.
3. Creating a new RSVP packet with a special object, in which the control information is carried by the special object being defined.

After identifying a control packet, a node can handle it using either an in-band or out-of-band approach. In the in-band approach, control packet clings together with data packets and receives the same level of forwarding treatment as other data packets, thereby it is subjected to being delayed or even dropped when the node is congested. This approach simplifies the design of the interior node. By examining the arrival of the control packets, one can also obtain a sample of the current congestion level of the forwarding path.

For the out-of-band approach, control packets receive special service, usually better than data packets, at an interior node. It requires a special arrangement within the forwarding module of an interior node. However, for a control algorithm that is sensitive to the round-trip-time and integrity of the control packet, the out-of-band approach is more appropriate.

3.2.3 Control Information

Various types of information can be carried in a control packet. However, the choice of type of information affects the capability of the ATC at the ingress node, and therefore, determines the controllability of the overall mechanism. The type of information can be categorised in terms of several attributes:

- *Type of indicator*
  This refers to what information is collected from interior nodes. It can be as simple as a binary *flag* which indicates congestion occurs, an *instantaneous or average buffer level or measured load*, or a more complicated measure of *higher order statistics*, e.g., buffer growth rate, rate of change of total load, etc.
- *Type of feedback*
  This refers to what information is returned to the ingress node. It can be in terms of binary *flag*(s), *explicit rate* or a form of *credit/token*.
- *Granularity*
  This refers not only to how coarse a measurement is done, e.g., *per-PHB-class*, *per-PHB* or *per-port*, but it also specifies how frequently a measurement is performed.
- *Directionality*

Direction here refers to how information is collected. Typically, information is collected in a forward direction where the control packet travels from an ingress node towards an egress node. In some cases, it can also be gathered in the reverse direction or even in both directions. However, it should be noticed that the forward and backward paths could be different depending on the routing protocol.

To select a type of information, network providers may consider the required controllability and processing capability of their network nodes. For other network management purposes, some routers may also have the capability of monitoring their loading condition. These loading statistics can also be used as a form of network information for this control purpose.

3.2.4 Adaptive Traffic Conditioner

Generally, the objectives of the adaptive traffic conditioning are to ensure that under any network loading conditions: (1) the traffic entering a DS domain conforms to the rules specified in the TCA; (2) the conditioned traffic will have "fair" share of the available resource inside a DS domain; (3) congestion can be effectively removed; and (4) resources within a domain are being utilised efficiently. In Section 3.1.3, we have described the general rules of the ATC functions. One way to realise these rules and objectives is to enhance the conventional TC with a *supplementary traffic profile*. Originally, the traffic profile is specified in a TCA and therefore is static in the sense that will not change over time or with network dynamics. The supplementary profile, however, is a profile derived from the original one and will be updated according to the state of the domain under control.

As in conventional TC, the actions taken on out-of-supplementary-profile packets may include delaying those packets until they become in-of-supplementary-profile (i.e. shaping), discarding those packets or re-marking the DS field of the packets to a particular codepoint. Since the supplementary traffic profile changes with the network dynamics, transient effects on these actions should carefully be handled. The following discusses these effects.

1. Dropping
   Notice that a change of traffic profile will trigger a change of dropping threshold. For aggregated TCP flows, an abrupt change in dropping level may cause many packets to be dropped at the same time. Eventually, it may trigger all TCP sources to back off and results in a poor overall throughput. To remedy this global synchronisation problem, one should avoid this "hard-limit" dropping.
2. Shaping
   When the traffic profile changes, not only should the output rate of the shaper be adjusted, but the size of the shaping buffer should also be updated. Again, if the adjustment causes the shaping buffer to be overflowed, the problem of global synchronisation should be avoided.
3. Marking
   A marker can adapt for the change of traffic profile in two possible alternatives:
   (1) Packets are promoted or demoted to other PHB within the same class; and
   (2) Packets are re-directed to another PHB class. Note that this may cause packets to be re-ordered.

3.2.5 Control Algorithm

In general, the control algorithm comprises two major components: *fair share computation* and *adaptation algorithm*. The fair share computation first calculates a target fair share value for each traffic flow. The adaptation algorithm then computes a feedback quantity such that the target fair share can be enforced at the ingress node.
Many algorithms are possible, but one can characterise and evaluate their performance by the following attributes:

- Fairness criteria: min-max fairness, proportional fairness or worst-case fairness
- Computational complexity: the amount of computation required and its relationship with the number of flows.
- Stability and convergence time: the time required reaching a target value, if possible.
- Capability to handle transient periods

4 An Instance of FC-DS

Note that this section is provided for clarification of concepts and for illustration of the significance of the feedback control extension. It is not intended to depict specific implementations or implementation requirements.

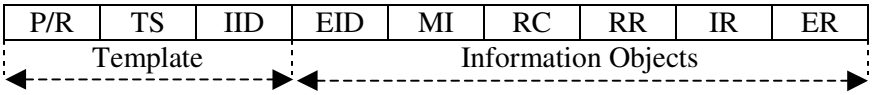
4.1 System Configurations

Table 2 summarises the system configurations that we have chosen for our control mechanism. The choices of the configurations are largely based on our initial experience and consideration. Detailed rationale for some configurations is discussed in [THESIS].

Functionality		Configurations
Overall	Control	Intra-domain control
	Fairness Criterion	Proportional fairness
ATC	Traffic Profile	Token bucket based
	Shaping	Adaptive with proportional buffer size
	Dropping	Adaptive with *soft random discard*
	Marking	Remarking ONLY within a PHB class
Probing/ Reporting	Info type (Probe)	Per-PHB-class based, averaged measured load (Exponential averaging)
	Info type (Report)	Explicit rate feedback
	Temporal Resolution	Time based periodicity
	Spatial Resolution	Per-aggregated flow based
	Identification/ Forwarding	CF-DSCP / CF-PHB (CF: Control Forwarding)
	Data Collection	Single pass, forward direction

Table 2 : Summary of System Configurations

Figure 4.1.1 depicts our proposed format for a control packet. Excluding the packet header, it is composed of two parts. The template part consists of information fields that are common to all possible mechanisms while the information objects part contains all vendor-specific fields.



P/R	Probe / Report Indication	MI	Measurement Interval
IID	Ingress node Identifier	RC	PHB Class of the referenced flow
EID	Egress node Identifier	RR	Requested Rate/profile
TS	Time-stamp/sequence no	IR	Ingress Rate/profile
		ER	Explicit Rate/profile

Figure 4.1.1 : Control Packet Format (Data area only)

4.2 Operational Details

The operational procedures of our control mechanism are as follow:

1. At the FC-DS boundary, the ingress node periodically samples its incoming flows. For each sampling interval, it generates and delivers a probe packet along with the data packets per aggregated flow. This probe packet carries the same header information as the sampled data packet, but it is remarked at the DS-byte with the CF-DSCP. The data area of the probe packet is filled with the information of this flow.
2. At any node inside a FC-DS domain, upon receiving a packet with CF-DSCP, the node first computes a suggested explicit rate using the information carried at the probe packet and its control algorithm. If the suggested explicit rate is smaller than the one carried at the ER-field of the received probe packet, the ER-field of the packet will be replaced. The updated probe packet is then forwarded to the next node.
3. When a probe packet is received by an egress node, a report packet is created and returned to the ingress node indicated by the IID-field of the probe packet. The report packet is identical to the received probe packet with exception of its P/R- and TS-field being updated accordingly.
4. Finally, when a report packet reaches the ingress node, the parameters of its corresponding ATC is updated. To remedy the global synchronisation problem in TCP flows, we introduce a mechanism called *soft random discard*. Figure 4.2.1 illustrates an adaptive traffic profiler with soft random discard.

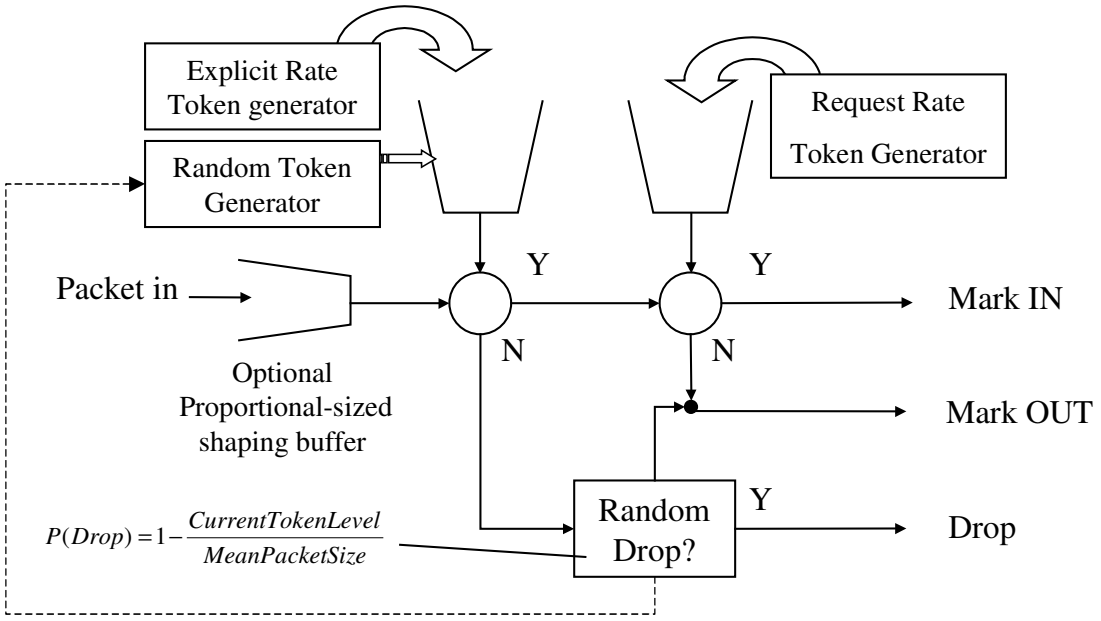


Figure 4.2.1: Adaptive Traffic Profiler with soft random discard

4.3 Performance Evaluation

To evaluate the performance of our chosen control mechanism, we conducted several sets of simulations. In this document, we show only some selected results. A complete report on the proposed system can be found in [THESIS].

4.3.1 NS-2 simulator implementation model

Figure 4.3.1 depicts an implementation of a FC-DS capable interior node. Note that the components of PHB Classifier, packet queues with various types of queue management schemes and output scheduler are commonly found in most DS nodes. For a FC-enabled node, a control module, which is tightly coupled with a load estimator and a collection of per-queue measurement modules, is included. In our design of a packet queue, the *Queue/RIO+* implements an AF PHB class with four drop preferences. The four drop preferences, which represent the packet attributes of IN/OUT-of-profile and UDP/TCP, can be ranked according to their dropping probabilities as IN-TCP < IN-UDP < OUT-TCP < OUT-UDP. In addition, the outputs of packet queues are controlled by a *Queue/PQ+* scheduler. *Queue/PQ+* is a simple rate-limited priority queuing that schedules packet delivery according to a pre-defined priority configuration. Furthermore, for the boundary nodes, an additional adaptive traffic profiler and a simple acknowledgement module are included in an ingress node and egress node, respectively.

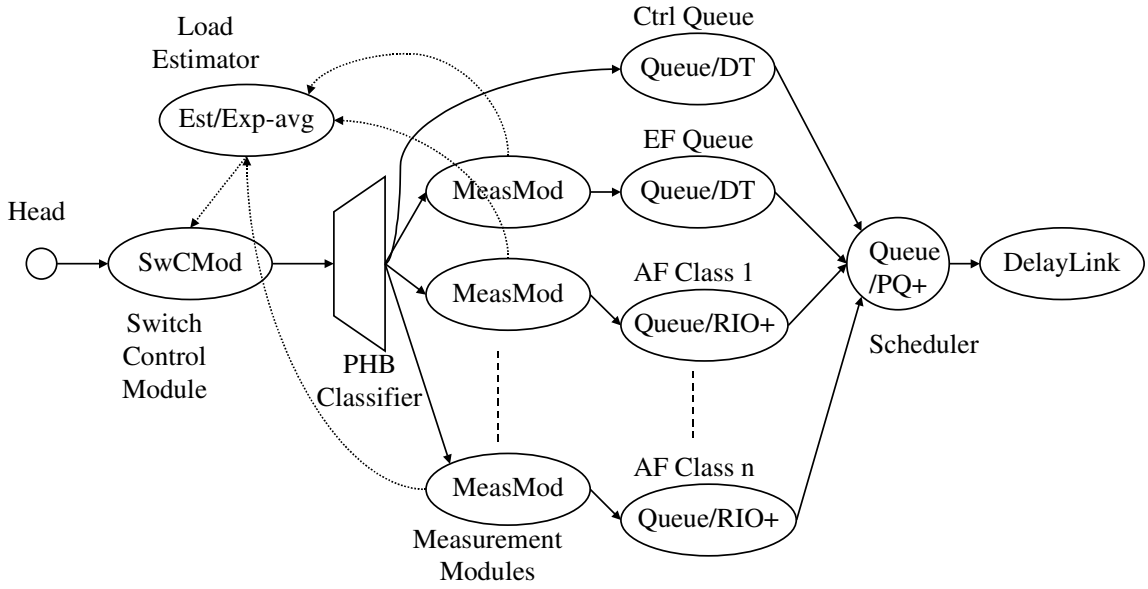


Figure 4.3.1: NS2 implementation model of a FC-DS Interior node

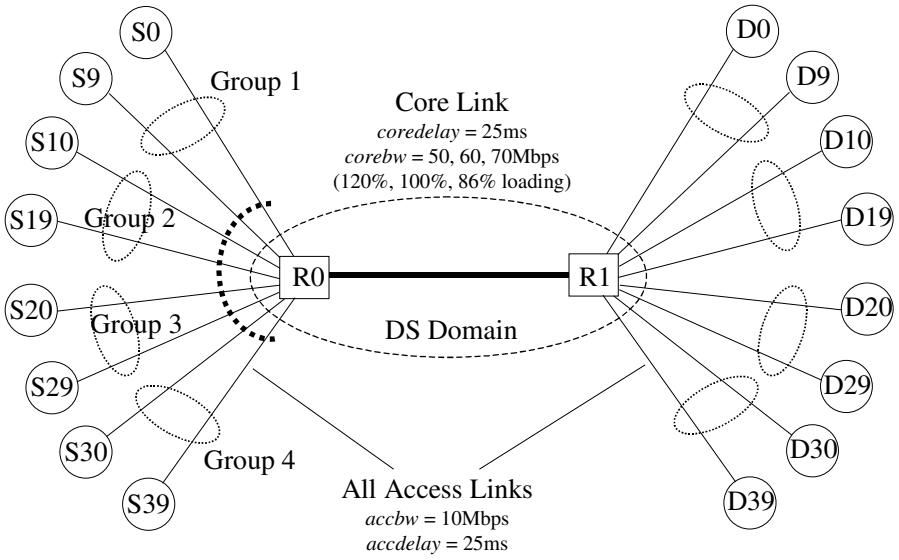
4.3.2 Selected Results & Discussions

Three different types of network topology are investigated, as shown in the following sections. Throughout the simulations, there are two types of flows, TCP and non-adaptive UDP flows, each of which carries different types of traffic. While the UDP flows carry the traffic generated by CBR sources, the TCP connections are all infinite sources that simulate FTP applications. The TCP agent implements either Reno-TCP or Sack-TCP. Moreover, all sources, both CBR and FTP, are randomly started with starting times uniformly distributed within the first second of the simulation time. Unless otherwise specified, all flows are sent using AF PHB. All data packets are fixed size, 576 bytes long.

Furthermore, we chose the following parameters throughout all simulation scenarios:

Parameters		Settings
Delay of an access link		Uniformly distributed between [0, *accdelay*]
Maximum queue size of all links		*Bandwidth* × *average RTT*
RIO+ (*minth/maxth/maxp*)	OUT	0.5·*maxQsize*/0.9·*maxQsize*/0.033
	IN	0.8·*maxQsize*/*maxQsize*/0.011
Profiler token bucket	CBR flows	2 × *packet size*
	TCP flows	*Requested rate* x *RTT*

Notation: FC-*x*-*y* : Feedback Controlled microflow *x* within aggregate *y*
UC-*x*-y : Un-Controlled microflow *x* within aggregate *y*
FC-S- : Feedback Controlled with adaptive shaper



Set	Group	Src	Dst	Src Type	Src Rate (Mb/s)	Request Profile (Mb/s)
1	1	S0-S9	D0-D9	CBR	@ 3.0	@ 1.0
	2	S10-S19	D10-D19	CBR	@ 3.0	@ 2.0
	3	S20-S29	D20-D29	TCP	/	@ 1.0
	4	S30-S39	D30-D39	TCP	/	@ 2.0
2	1	S0-S9	D0-D9	TCP	/	@ 0.5
	2	S10-S19	D10-D19	TCP	/	@ 1.0
	3	S20-S29	D20-D29	TCP	/	@ 1.5
	4	S30-S39	D30-D39	TCP	/	@ 2.0

Figure 4.3.2: Topology 1 - Single congested link topology

4.3.2.1 Effect of non-adaptive flows

In the presence of non-adaptive flows, all TCP connections are degraded, even though they are protected inside their requested profile envelope as long as the network has been adequately provisioned. However, excess bandwidth or any scarce resource during congestion is taken by non-adaptive flows because the TCP sources back off when their OUT packets are dropped. As indicated from Figure 4.3.3, this unfair situation can be remedied by employing a feedback control mechanism. In a FC-DS domain, non-adaptive flows are regulated according to fairness criterion such that they are prevented from monopolising the available resource.

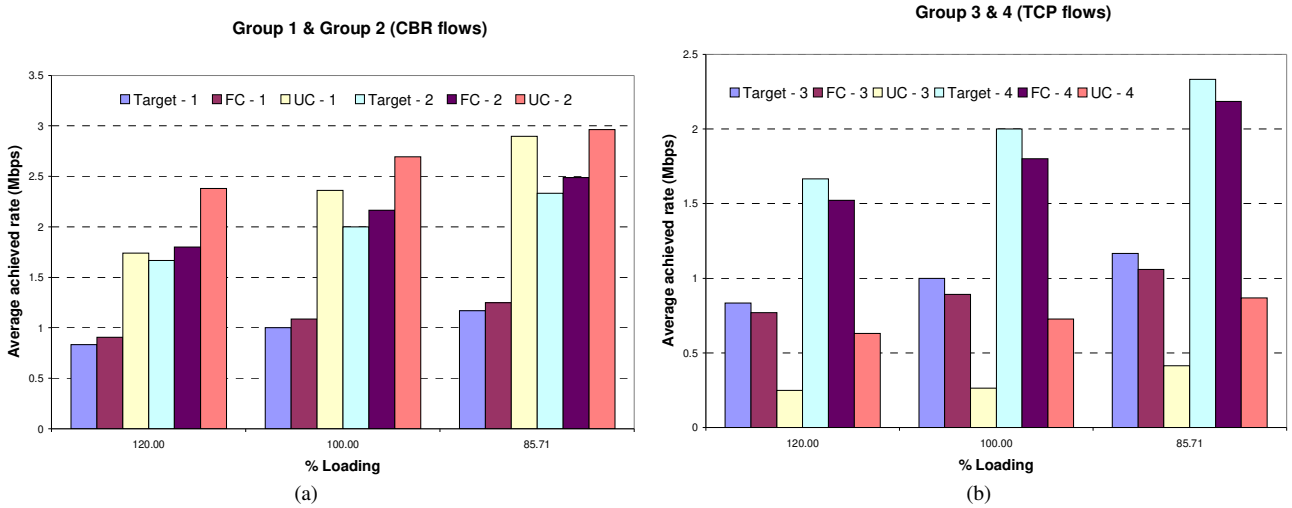


Figure 4.3.3: Average achieved rate (Set # 1)

4.3.2.2 Effect of requested profiles

From Figure 4.3.4, we notice that connections with small requested profiles reach or exceed their profiles noticeably in conventional DS. This is due to the variation of TCP congestion window. After the window is closed because of packet losses, the connections with small requested profile return to their legitimate window size quicker than those with larger profiles, thus they can compete for the excess bandwidth sooner. In FC-DS, since the supplementary traffic profile opens gradually in a fair manner, it, in effect, provides a fair ground for flows with different requested profile to compete for the available resource. Hence the percentage error deviated from the target fair-share rate is significantly improved.

4.3.2.3 Effect of inter-class interference

To study the influence of inter-class interference, we have repeated the simulation set#1 with an additional connection that injects an interfering traffic of 20Mbps CBR flow from 20s to 40s using the EF-codepoint. Since traffic on EF-PHB has a higher forwarding priority than AF classes, it creates a sudden starvation of resource. While the uncontrolled flows completely fall away from the target rates, it is noticed from Figure 4.3.5 that the feedback controlled flows follow closely with the abrupt change of available bandwidth. It also shows that the proposed control mechanism is free from stability problems during the transition period.

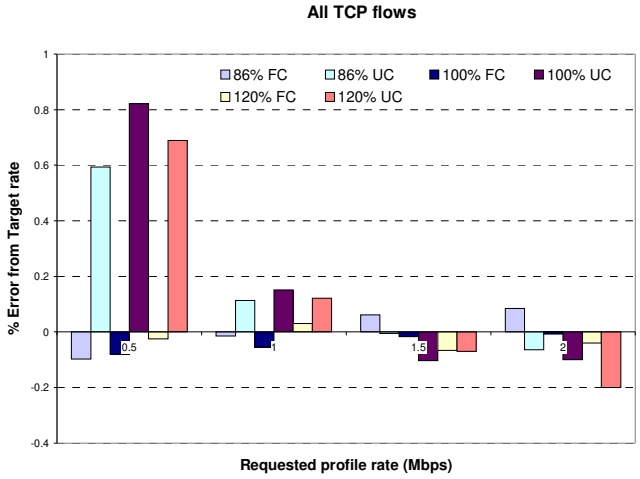


Figure 4.3.4: Average achieved rate (Set # 2)

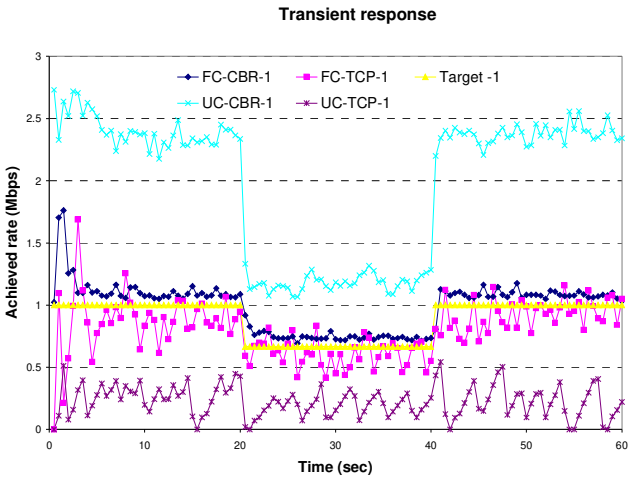
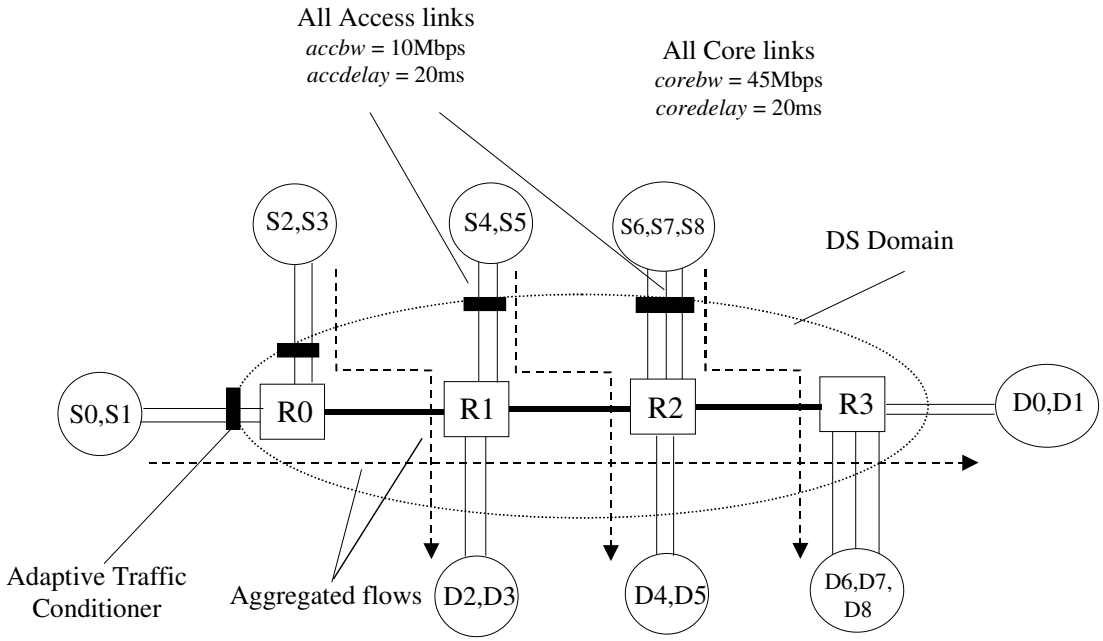


Figure 4.3.5: Time response of average achieved rate (Set # 3)

4.3.2.4 Effect on longest flows



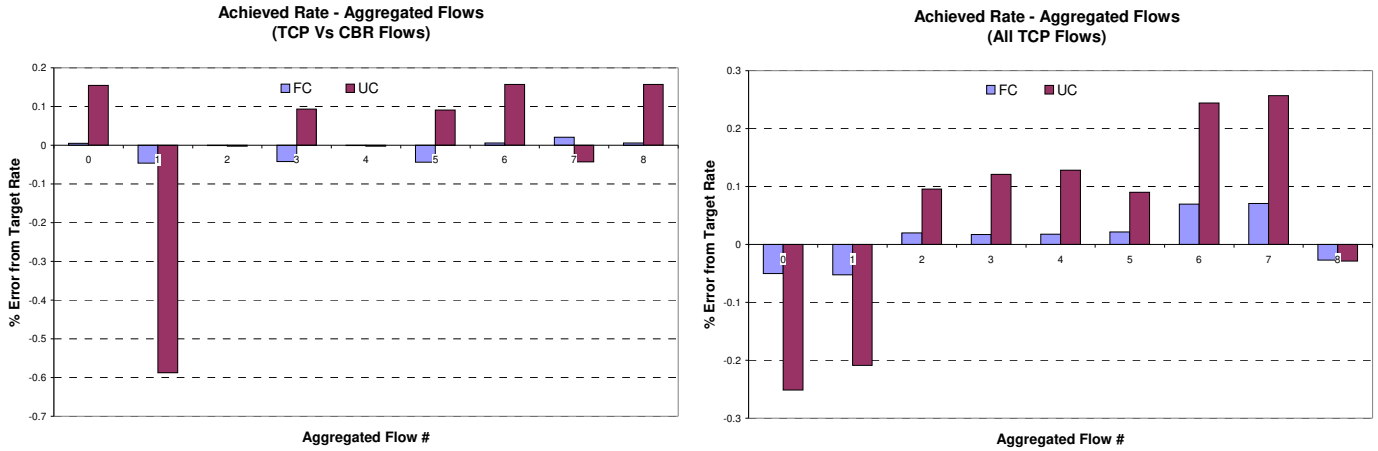
Set	**Agg. Flow	Src	Dst	Src Type	Src Rate (Mb/s)	Requested Profile (Mb/s)
4	0	S0	D0	CBR	1.0	10.0
	1	S1	D1	TCP	/	10.0
	2	S2	D2	CBR	1.0	10.0
	3	S3	D3	TCP	/	10.0
	4	S4	D4	CBR	1.0	10.0
	5	S5	D5	TCP	/	10.0
	6	S6	D6	CBR	1.0	10.0
	7	S7	D7	TCP	/	10.0
	8	S8	D8	CBR	2.0	20.0
5	0 - 7	S0 - S7	D0 - D7	TCP	/	@ 10.0
	8	S8	D8	TCP	/	20.0
6	0 - 7	S0 - S7	D0 - D7	CBR	@ 2.0	@ 10.0
	8	S8	D8	CBR	2.0	20.0

***** Each aggregated flow contains 10 microflows.***

Figure 4.3.6: Topology 2 - Multiple congested links topology

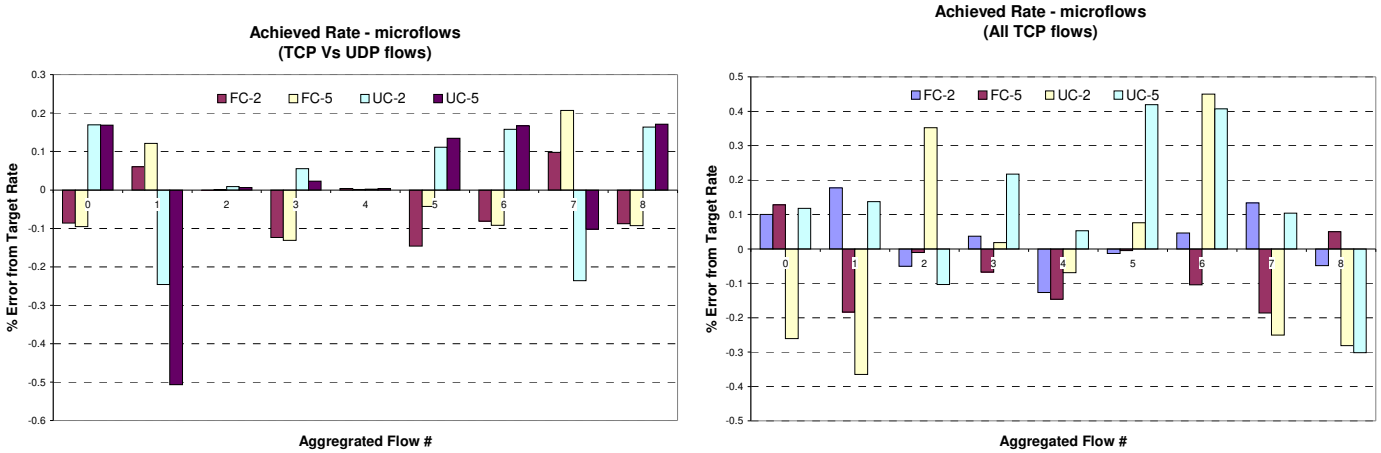
A *long flow* refers to a flow that traverses a number of nodes. In topology 2, microflows within aggregated flow-0 and flow-1 are the longest flows. In conventional DS, long flows usually have poorer performance than other flows. This is because every time a packet enters a node, it has to compete with others for available resources. Since packets or flows are indistinguishable inside the core of a DS domain, the more the number of nodes they travel, the higher the probability that they will experience loss or delay. In FC-DS, long flows are being protected by regulating access of *short flows* such that a fairer sharing of resource is maintained. Figure 4.3.7, Figure 4.3.8 & Figure 4.3.9 confirm that the achievable rate and delay of the long flows can be improved significantly under FC-DS.

Another interesting observation is that FC also helps improving the performance of short flows under certain circumstances. For topology 2, congestion occurs at the last hop between R2 and R3, i.e., severe packet dropping occurs at R3 while excess resources are available at other nodes. In an uncontrolled environment, since S0 is non-adaptive and not aware of any congestion at the downstream nodes, it continuously injects packets into the domain. These packets maintain a certain level of buffer occupancy at router R0, R1 and R2 even though they are eventually dropped at R3. Under this situation, not only network resources are wasted at the non-congested nodes, but other flows are also prevented from accessing the originally available resources. By introducing a FC at the DS edge, packets from S0 can be throttled earlier at R0 and thereby, it preventing the DS domain from being persistently congested. Figure 4.3.7, Figure 4.3.8 & Figure 4.3.10 confirm this observation.



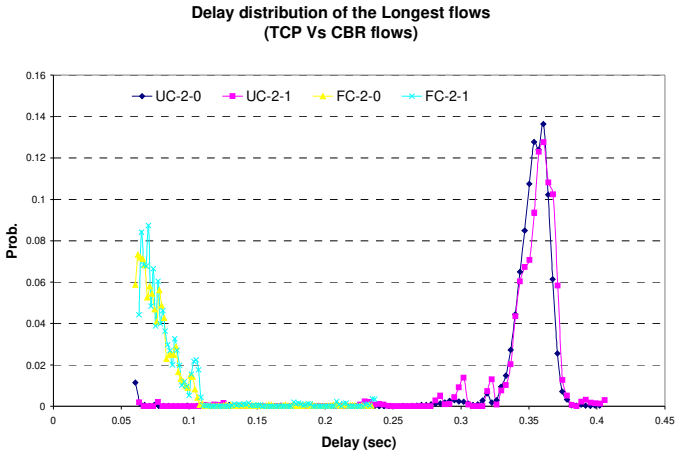
(a) Set#4 (b) Set#5

Figure 4.3.7: Achieved rates for aggregated flows

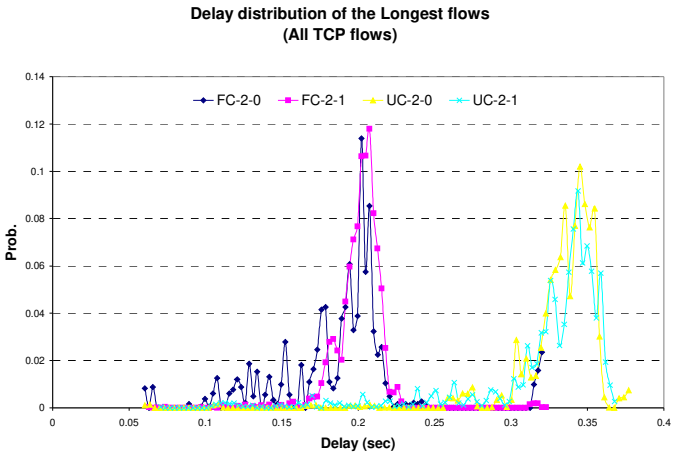


(a) Set#4 (b) Set#5

Figure 4.3.8: Achieved rate for microflows-2 & 5 within an aggregate

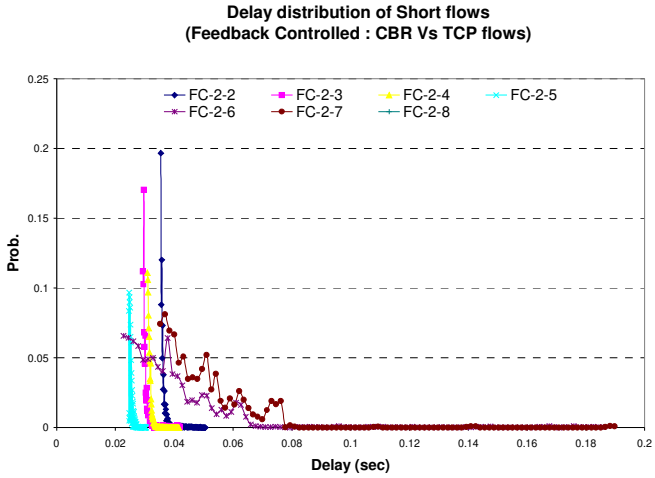


(a) Set#4

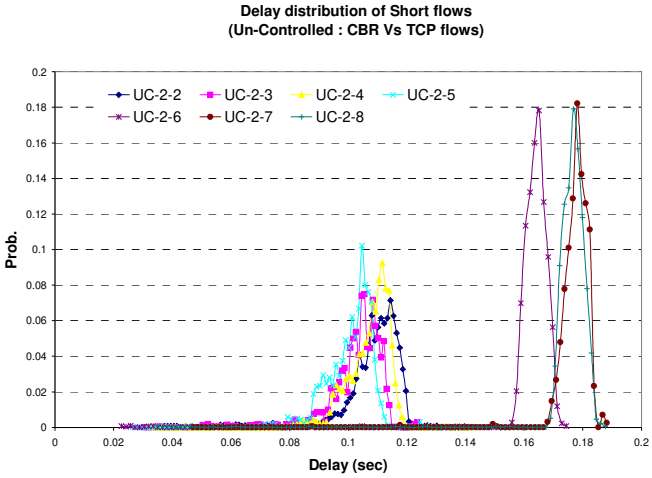


(b) Set#5

Figure 4.3.9: Delay distribution for long microflows-2 of aggregate-0 & 1



(a)

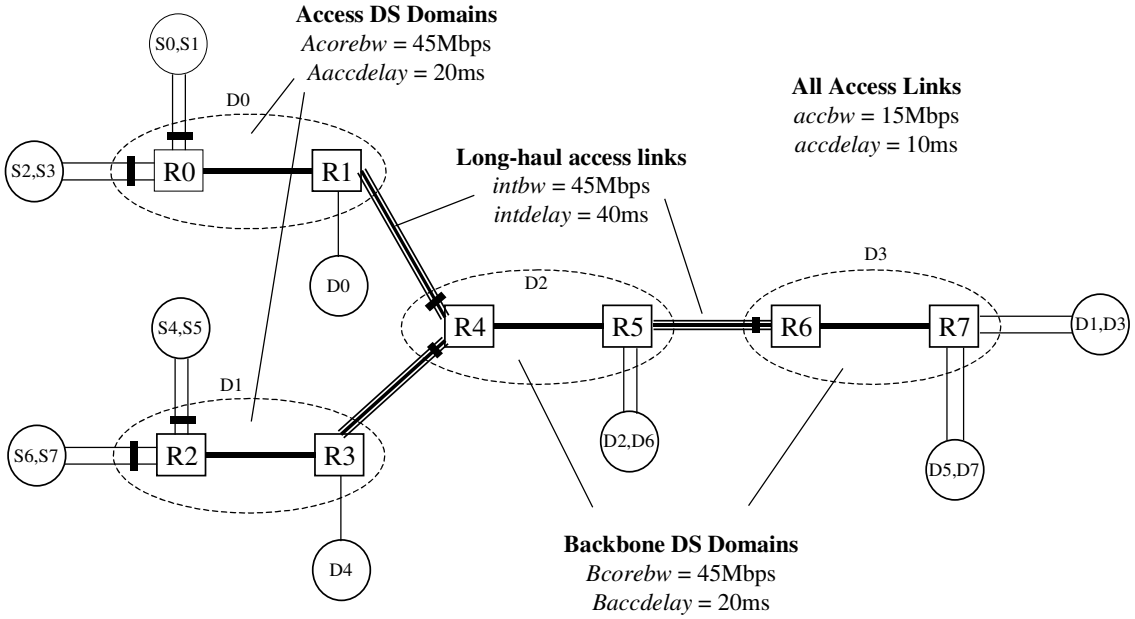


(b)

Figure 4.3.10: Delay distribution of short microflows-2 within an aggregate (Set # 4)

4.3.2.5 Effect of Round-Trip-Times

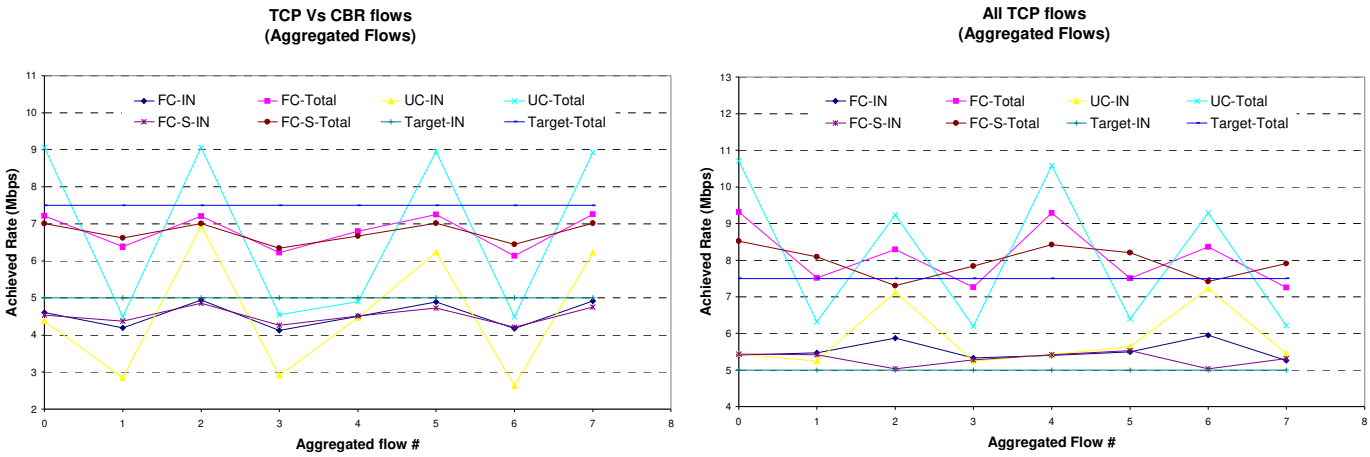
Figure 4.3.12 & Figure 4.3.13 show the performance of flows under a typical multiple-tier scenario as in topology 3. Besides the long flow effect mentioned earlier, the influence of RTT on the achieved rate is also noticeable. For the case without FC, it is observed that some connections do not achieve their target fair-share rates, while others severely exceed their targets. In the results of Set#8, flow-0 and flow-4, which have the shortest RTTs, grow their congestion window more quickly and come out of their requested profile envelopes more frequently to exploit excess bandwidth using their OUT packets. However, the OUT packets cannot prevent the IN packets of other flows from entering the router queue. Therefore, flows with larger RTTs are at least assured of their requested profile rates, but they can hardly receive a fair share of excess bandwidth. Again, with the feedback control mechanism, this effect can be effectively removed.



Set	**Agg. Flow	Src	Dst	Src Type	Src Rate (Mb/s)	User requested profiles	Domain Requested Profiles (Mb/s)	
							D2	D3
7	0	S0	D0	CBR	1.0	5	15	20
	1	S1	D1	TCP	/	5		
	2	S2	D2	CBR	1.0	5		
	3	S3	D3	TCP	/	5		
	4	S4	D4	TCP	/	5	15	
	5	S5	D5	CBR	1.0	5		
	6	S6	D6	TCP	/	5		
	7	S7	D7	CBR	1.0	5		
8	0 - 7	S0 - S7	D0 - D7	TCP	/	@ 5	@ 15	20

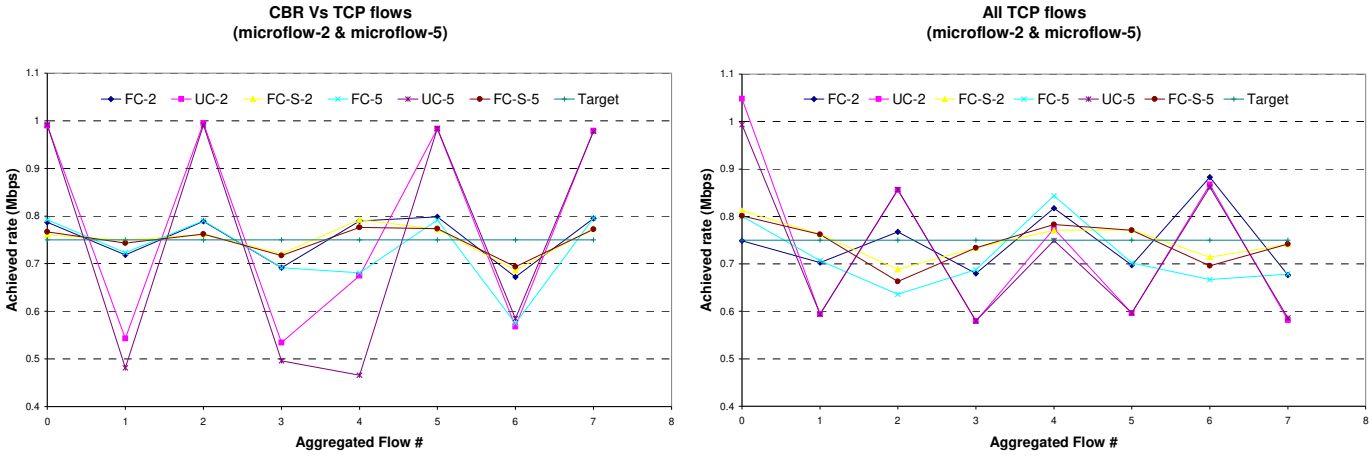
***** Each aggregated flow contains 10 microflows.***

Figure 4.3.11: Topology 3 - Multiple-tier topology



(a) Set#7 (b) Set#8

Figure 4.3.12: Achieved rate for aggregated flows

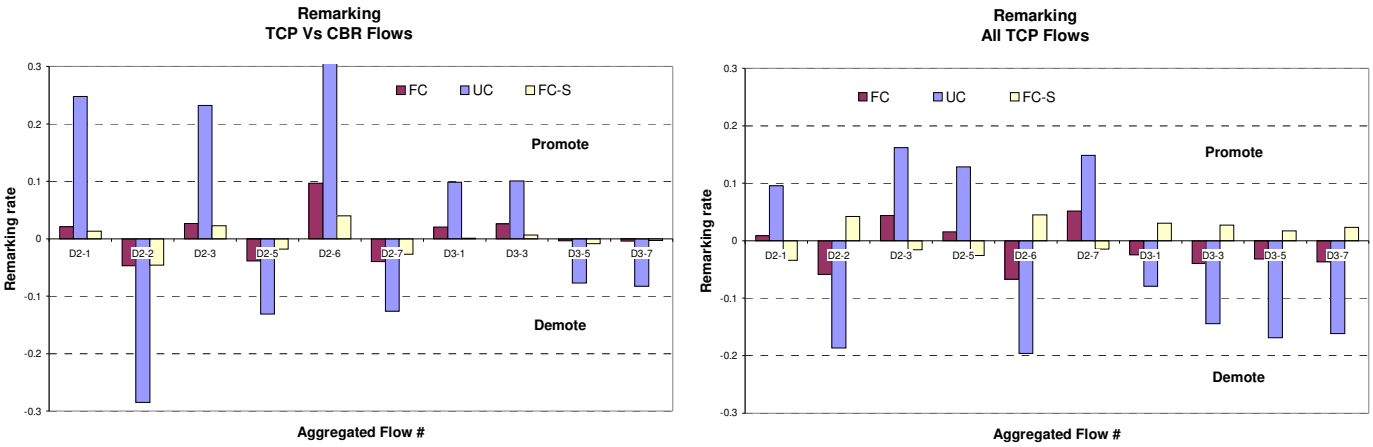


(a) Set#7 (b) Set#8

Figure 4.3.13: Achieved rate for microflows

4.3.2.6 Effect on remarking rate

At each merge point, traffics are aggregated and therefore, traffic-bursts can be accumulated throughout the network. When a traffic-burst hits the edge of a DS domain, it is remarked according to the contracted inter-domain aggregated profiles. Hence the higher the remarking rate, the more bursty the incoming traffic is. Figure 4.3.14 illustrates the remarking rate of different flows at the edges of domain 2 and 3. We observe from Figure 4.3.14(b) that with the absence of a feedback control mechanism, traffic tends to be more bursty at the edge of a domain even though all traffics are of the same type. Moreover, it is noticed that remarking occurs at an unfair fashion upon different aggregated components. This implies traffic is highly unbalanced at the merge point. In essence, while a feedback control mechanism can help reduce the traffic burstiness, it can also balance the composition of the aggregated traffic.



(a) Set#7 (b) Set#8

Figure 4.3.14: Remarking rate for aggregated flows
(Notation: D*x*-*y* = Domain *x* - aggregated flow *y*)

In summary, this Section presents an example control mechanism derived from the general paradigm. It has been shown that the overall feedback controlled DS can offer a better performance and controllability over the conventional DS. This specified mechanism is by no means the only possible way to perform feedback control. Possible specification of other control mechanisms is left for future study.

5 Other Considerations

5.1 Standardisation

Since our goal is to enable network providers to implement their own control mechanisms according to their need and policies, the requirements to be standardised should be kept minimal. We suggest standardising only the following:

(1) Extended functional requirements for architectural components given in Section 3.1, and

(2) Probe/report (control packet) format: This includes only the template part of the control packet and one of the identification methods suggested in Section 3.2.2. If consensus is to use the out-of-band approach with a special DSCP, a DS codepoint assignment is required.

5.2 Inter-domain control

So far, we have assumed an intra-domain control mechanism. In some cases, an inter-domain control may be preferable. Usually, domains are operated by different network providers. To enforce a global control mechanism across multiple DS domains, several problems need to be resolved.

(1) Policy conflicts: different providers usually maintain their own policies in terms of management objectives, network provisioning, etc. To resolve any potential conflicts, we suggest that the TCA between two domain operators should be augmented with *a domain control agreement* (DCA).

(2) Compatibility among different control mechanisms: if control packets are not terminated at the boundary of a domain, the control algorithms and information models used in different domains need to be compatible. Otherwise, a domain-to-domain control is not possible.

(3) Longer control packet RTT: since control packets need to traverse more than one domain, a longer round-trip control delay is unavoidable. The overall adaptation algorithm should take this into consideration.

5.3 Interoperability with non-feedback-control-extended DS components

We define a non-feedback-control-capable node (non-FC-capable) as a node which does not interpret control packets (probe / report) and / or does not implement some or all of the functions mentioned in Section 3.1. Although details of the control mechanism may vary, generally, in order to obtain a consistent domain control, all boundary nodes must be upgraded to feedback-control capable nodes.

Inside the DS domain, the non-FC capable interior nodes are required to maintain basic forwarding treatment for the control packet. However, it is desirable that they should have enough resources such that they will never become bottleneck points.

5.4 Multicast

Note that the issue of multicast is still an active research topic in Diff-serv WG. In order to control multicast traffic in the context of FC-DS, one fundamental requirement is to duplicate the probe packet at the point of divergence. At the ingress node, when multiple reports are returned from the leaf nodes of the multicast tree, an algorithm is required to consolidate the reports and derive a suitable set of ATC parameters. Details of these issues need further study.

5.5 Security

We only discuss security issues in the context of the control mechanism. There are two issues of protection involved:

(1) Protection upon control packets: this mainly refers to the integrity and privacy of the information carried inside the control packet. A FC-DS node should always prevent any control packet from being intercepted, modified illegally or read without authorisation.
(2) Protection against forged control packet attack: A FC-DS boundary node should have a strategy to identify forged control packet and prevent its operation from being affected.

Details of these protection strategies and other security concerns need further study.

6 Summary

This draft proposes an extension to DS that enable a feedback control mechanism to be implemented on a DS domain. With the feedback control mechanism, network providers can manage their traffic more effectively, thereby achieving a better resource sharing and more efficient resource utilisation. A control mechanism, which is akin to ABR service in ATM, can be derived from our proposed extension. However, it is more flexible than ABR in terms of functionality and in particular, it is tailored for network-layer instead of link-layer or transport-layer of the protocol stack.

This document is intended to serve as a starting point for the discussion in this direction. The next version, if the WG requests, will further specify the detailed requirements.

Acknowledgement

This work is supported in part by research grant from CITR, but that the views are the authors'.

References

[DSARCH] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[DSFMWK] Y. Bernet, J. Binder, S. Blake, M. Carlson, S. Keshav, E. Davies, B. Ohlman, D. Verma, Z. Wang, W. Weiss, "A Framework for Differentiated Services", IETF Internet Draft <draft-ietf-diffserv-framework-01.txt>, October, 1998.

[DSHEAD] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", IETF RFC 2474, December 1998.

[DSBOUND] Y. Bernet, D. Durham, F. Reichmeyer, "Requirements of Diff-serv Boundary Routers", IETF Internet Draft <draft-bernet-diffedge-01.txt>, November 1998.

[RVCTRL] B. Ohlman, P. Koskelainen, "Receiver control in Differentiated services", IETF Internet Draft <draft-ohlman-receiver-ctrl-diff-01.txt>, September 1998.

[ASIN] J. Ibanez, K. Nichols, "Preliminary Simulation Evaluation of an Assured Service", IETF Internet Draft <draft-ibanez-diffserv-assured-eval-00.txt>, August 1998.

[ASKLT] H. Kim, W. Leland, S. Thomson, "Evaluation of Bandwidth Assurance Service using RED for Internet Service Differentiation", Pre-print, ftp://ftp.bellcore.com/pub/world/hkim/assured.ps.Z

[ASBW] A. Basu, Z. Wang, "A Comparative Study of Schemes for Differentiated Services", *Bell labs Technical Report*, August 1998.

[NTIMP] M. Biegi, R. Jennings, S. Rao, D. Verma, "Supporting Service Level Agreements using Differentiated Services", IETF Internet Draft <draft-verma-diffserv-ntimplem-00.txt>, November 1998.

[THESIS] H. Chow, On Supporting QoS over the Internet, *PhD Dissertation, University of Toronto,* work in progress.

[FENG] W. Feng, D. Kandlur, D. Saha, K. Shin, "Adaptive Packet Marking for Providing Differentiated Services in the Internet", in proc. of Int. Conf. on Network Protocols, October 1998.

Authors' Addresses

Hungkei (Keith) Chow
Email: keith@nal.utoronto.ca
http://www.comm.utoronto.ca/~keith

Alberto Leon-Garcia
Email: alg@nal.utoronto.ca
http://www.comm.utoronto.ca/~alg

Network Architecture Laboratory
Dept. of Electrical & Computer Engineering
University of Toronto
10 King's College Road,
Toronto, ON, M5S 1G4,
Canada