Shared Control of Networks using Re-feedback; An Outline

Bob Briscoe Sébastien Cazalet Andrea Soppera
Arnaud Jacquet
<{rbriscoe|scazalet|asoppera|ajacquet}@jungle.bt.co.uk>
BT Research,
B54/130, Adastral Park, Martlesham Heath, Ipswich, IP5 3RE, UK

Abstract

Properly characterising paths is an important foundation for resource sharing and routing in packet networks. We realign metrics so that fields in packet headers characterise the path downstream of any point, rather than upstream. Then closed loop control is possible for either end-points or network nodes. We show how incentives can be arranged to ensure that honest reporting and responsible behaviour will be the dominant strategies of selfish parties, even for short flows. This opens the way for solutions to a number of problems we encounter in data networking, such as congestion control, routing and denial of service.

1 Introduction

This paper introduces an improved pattern for feedback in networks with unidirectional data flows, such as packet networks like the Internet. Characterisation of paths through networks is the foundation of two of the hardest problems in networking: resource sharing and routing. Information about the path can be collected in header fields as data traverses a path, e.g. time to live (TTL) or congestion notification (ECN [6]). Previously, as each node characterised its local section of the path, the header values accumulated upstream path knowledge. By realigning feedback, we arrange each field to characterise the remaining *downstream* path. The re-alignment could not be simpler. We aim to reach a target for the metric at the destination, rather than aligning the origin at the source. The destination feeds back any error to the source, which adjusts the initial values in the next packets it sends, aiming to zero the error at the destination. At flow start, before any feedback, packets are flagged to indicate that their metric(s) are uncertain. Each unflagged packet then carries with it the state of its downstream path, albeit a round trip ago. As each packet arrives, it gives every relay a view of its downstream path. Wherever there is a need to distinguish, we term this pattern 'receiver centred feedback' or 're-feedback'[1] for short.

[1] In hindsight, source-aligned feedback should have had the different name, as re-feedback is like classic feedback.

The essence of feedback is that *information* about the effect (the output) of a system is returned to the *cause* (the input) so that it may alter its behaviour to better achieve the desired effect. Taking congestion as an example, feedback of ECN from the output requires the source to reduce its load thus reducing ECN at the output and therefore congestion itself on that path. But, in a network, the output at the destination *can* be influenced by any node along a path, not just the source. Each node may influence path congestion by the route it chooses and the priority it gives to the flow (whether it recognises flows or not). Each node is a contributory *cause* of what happens downstream.

The current Internet architecture was built on the assumption that sources could be trusted to control congestion and routers could be trusted to control routing. As part of the effort to redefine the architecture, we propose that feedback should be designed for a tussle over who controls what [3], arranging for downstream path information to be equally available to any node. Ensuring information is carried back to *all* the causes of the effect, rather than just to the source cause, opens the way for natural solutions to a number of problems we encounter in data networking.

We seem to have to trust everyone in the feedback loop not to distort information about the the physical realities of a path for their own selfish ends. In the body of this paper (§5) we use congestion notification as a concrete example to show how we can arrange everyone's incentives to promote honest reporting *and* behaviour — despite the stakes being high when competing for scare resources. We also briefly discuss why re-feedback incentivises sociable behaviour during the first window of a flow, making a network robust even if traffic is dominated by short flows. We explain why re-feedback then becomes the first line of defence against denial of service in unicast datagram networks. We also briefly explain how re-feedback will provide a trustworthy and continuous source of routing information to all nodes (§4).

But first, we describe the basic re-feedback mechanism with more precision, before applying it to

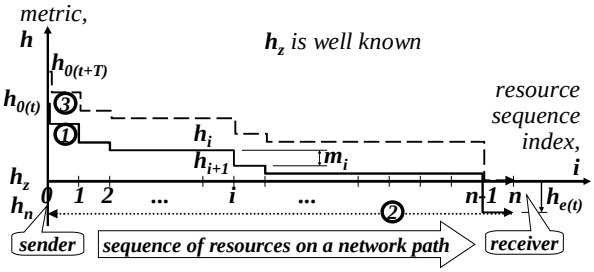

Figure 1: Re-feedback

congestion control and routing. After creating the incentive structures described above, we end by describing related work, outlining planned work and drawing conclusions.

2 Re-feedback

We will use re-feedback of congestion notification as a concrete example, because of its importance in controlling quality of service. In practice, path characterisation would consist of other metrics such as hop count or one-way propagation delay, which can be arranged in similar ways. For brevity, we take a mathematical approach to the values carried in data headers, deferring protocol engineering, such as single-bit marking, to a future publication.
Consider a path across a network consisting of a sequence of resources, i; $0 \leq i < n$. We place a metric, h, in a notional explicit congestion level (ECL) field in the network layer header of all data packets, intended to represent the *downstream* path congestion level. h_i is the value of the field before processing at the ith resource. The locally determined congestion level at the ith resource is m_i, which is subtracted[2] from the level of the metric in the packet. So, $h_{i+1} = h_i - m_i$. m_i is a function of the load at the ith resource. The RED algorithm [8] is an example of an approximation to this function, but, given our mathematical approach at this stage, we would use its internal marking probability (drawn from [0,1]) directly for m_i.
We will now consider the first of a flow of packets (step (1) in figure 1). The sender should estimate the initial value of ECL, h_0, to place in the packet and store this value. After transmission over the path, the ECL on leaving the $n-1$th and last resource will be $h_n = h_0 - \sum_{i=0}^{n-1} m_i$. The receiver then feeds back h_n to the sender using a relevant end to end protocol above the network layer (step (2)). When this feedback arrives at the sender, it can infer the path congestion level, $h_p = \sum_{i=0}^{n-1} m_i = h_n - h_0$. The sender may now adjust the rate at which it sends subsequent packets according to its congestion control algorithm. We arrange for a reference value of the congestion level to be well known (standardised) as the target congestion level to be aimed for by the time a packet reaches its destination. In this case $h_z = 0$. The sender calculates the 'error' $h_e = h_n - h_z$ and adjusts its estimate of the initial ECL to use for the next packet to send over this path to $h_{0(t+T)} = h_{0(t)} - h_{e(t)}$, to ensure that the ECL tends to reach h_z on subsequent arrival at the destination (step (3)).
For this next packet and all subsequent packets, if the path congestion h_p remains unchanged, $h_n = h_z$. However, if the path congestion changes, h_0 will reflect the change within one round trip, in order to ensure that the ECL field still tends to h_z at the destination. Thus, values of h_i at any point on any path always give a measure of congestion downstream of that point, albeit a round trip ago and further modulated by any changes on the upstream path just traversed by the packet. We use $\rho_i = (h_i - h_z)s$ to denote the *per packet* downstream path shadow price (DPSP) for a bit-congestible resource, where s is the packet size in bits.
The round trip delay for the ECL field is unavoidable, and no less than that the sender has always had to cope with when controlling load in response to feedback. When load or capacity is controlled at an intermediate node, rather than the sender, the minimum theoretical delay is the propagation delay from receiver back to intermediate node. Instead, by taking a full round trip, we achieve *near-continuous* path knowledge with slightly more delay. The theoretical minimum delay would require a stream of data reverse-routed to every one of the sequence of addresses on the downstream path.
Often, as a packet traverses every tunnel or logical link, the intermediate destination of the outer header may not be capable of generating fast feedback to the source. Rather, the node capable of fast feedback will be the destination of some inner network layer header. Thus, re-feedback will generally only be appropriate at layers where feedback is at a packet rate of the same order as the forward packet flow. However, re-feedback draws downstream path information to the head of a link or tunnel, so that appropriate congestion control or routing can be done *before* encapsulating a packet.

3 Shared congestion control

Controlling quality of service fundamentally concerns allocating resources during congestion. The seminal work of Kelly *et al* [12] proves that path shadow pricing provides the correct incentives for

[2] For brevity, we have removed background mathematical formality. For instance summing congestion along a path is an approximation only valid when congestion rates are small, which stems from a formal definition of path congestion as a product of probabilities. Also we have removed distinctions between per packet and per bit congestion.

rational networks and users to meet a wide spectrum of requirements in response to any pattern of resource congestion. The ECN mechanism employed is also simple, elegant and already in the process of standardisation. The whole point of *shadow* pricing is that it can control demand by transformation into other effects than market pricing, such as capacity or priority variation or flow admission control. However, because ECN accumulates towards the receiver, and is fed back above the network layer, it can be hidden from the sender's network operator, who is best placed to check the sender's behaviour but not the information that drives it.

The only reliable way round this is to charge the receiver a dynamic *market* price for received ECN, expecting some end to end settlement in order to incentivise the sender's response to congestion. Not only does this open receivers to 'denial of funds' attacks, it also requires end-customers to accept dynamic pricing. There is extensive evidence that most won't [13]. However, if we use re-feedback for congestion marking, the downstream path shadow price can be measured at any point throughout the network layer, most importantly at the ingress edge. It can then be used without conversion into a market price — as should be possible with a shadow price.

Four main ways have been proposed to share available capacity among users or flows: fairly, differentiated, by admission controlled reservation or by dynamic willingness to pay. Once downstream path characterisation is available to *both* sources and relays, we believe that each of these service models can be achieved under the control of any of:

- relays, dynamically prioritising or blocking each user's service, using shadow pricing internally to ensure priority or admission is fairly distributed;
- sources, but policed by network nodes to detect or prevent consumption in excess of differentiated or fair allocation;
- sources, but constrained to act responsibly by dynamic pricing.

The choice over which control model any one network offers will be controlled by greater forces in society at large, such as competitive markets or the regulatory policies of public agencies. We call this 'control over control', another way of saying it is designed for tussle [3].

Below we briefly outline examples of how re-feedback might help each service model to be realised, although they are merely initial ideas yet to be verified by experiment. We believe many more services can be created using re-feedback. For instance, another example of a weighted differentiated service is provided by Siris [15]. It relies on intercepting ECN feedback from the transport layer in order to respond to a combination of wireline and CDMA cell congestion along the path. Re-feedback would allow Siris's algorithm to work purely at the network layer.

In all cases, the general principle applies that a higher penalty must be paid in order to send a higher downstream path shadow price into the network. So, the more congested a path is, the greater the penalty. The penalty may be reduced priority, increased blocking probability or increased payment.

Fairness With re-feedback it would be possible for the sender's network operator to police the TCP compatibility of a source early enough in the path to prevent any unfair sharing of resources. Currently, with no characterisation of each downstream path at the network layer, it is hard for policers to maintain anything other than *rate* fairness. Schemes that penalise the highest rate flows, such as those derived from Floyd and Fall [7], cannot discriminate between real misbehaviour and flows which would genuinely be allocated a high rate — where slower flows would be unable to use more capacity due to congestion elsewhere or a slow feedback loop. With re-feedback, both path congestion rate and the time remaining to the destination could be carried in packet headers and their truth relied on (see §5). The policer would then have the necessary information to mirror the TCP rate algorithms.

Differentiated service gateway With re-feedback, we believe we can create a tiered service where higher traffic classes are much less affected by congestion than lower classes. It is much like Diffserv, but with three advantages: i) an ingress node can be responsive to congestion along the length of all its downstream paths, sharing out its congestion response between users and classes in a principled way; ii) the pattern of congestion notification throughout the network directs investment in aggregate capacity on a per interface basis in order to meet demand[3]; and iii) the relative weights between classes on each node should adjust automatically, with zero configuration. Without yet having simulated this idea, we cannot be certain of these theoretical claims. We are less certain of a potential fourth claim: iv) the capability may require no more than active queue management on all routers except those at the ingress edge of a multi-domain network. We show one of these ingress routers in figure 2.

Internally the router pays the congestion price required so that premium traffic need not respond

[3]New demand prediction would still be necessary.

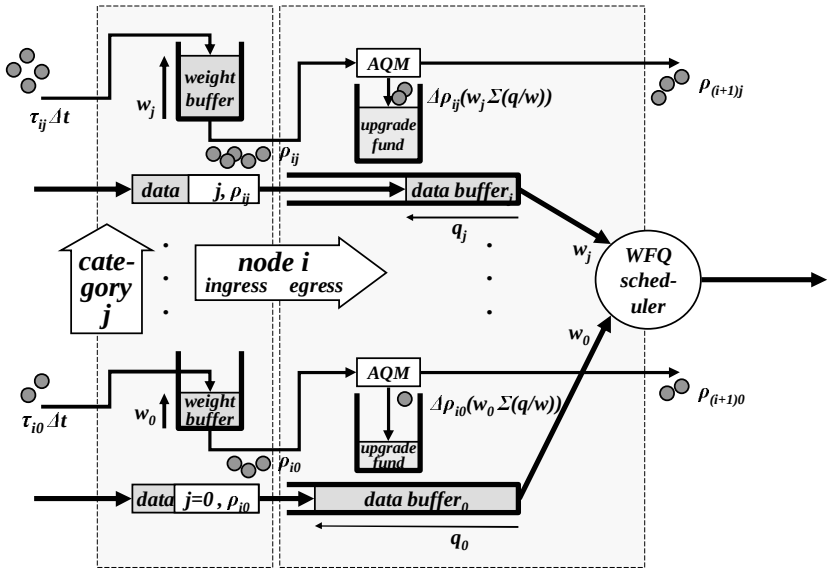

Figure 2: Re-feedback-based differentiated service node

to congestion. But, rather than charging the end-customer a dynamic price, it is charged internally to a weight buffer. The customer merely pays a regular subscription, which is modelled as a constant token fill rate into the same weight buffer. The difference between the constant fill rate, τ and the variable drain per packet ρ leads to a varying number of tokens in the buffer (the weight) over time. The end-points use re-feedback normally, inserting sufficient shadow price ρ into packets to cover the downstream path. So the more downstream congestion and the faster the data rate, the faster the weight buffer empties.

There is one weight buffer per category of traffic. A category may map to a traffic class (Diffserv code point) across many customers or, if the node is powerful enough, it may map to one traffic class for each customer. Traffic is scheduled using one of the practical approximations to weighted fair queuing. The varying depth of each buffer weights the scheduler for each category. Thus, service is prioritised proportionate to the token fill rate modulated by the shadow cost of congestion experienced on the downstream path.

Each packet is queued in a data buffer, where the local shadow price $\Delta\rho$ is subtracted dependent on the queue length. The more highly weighted queues apply a proportionately higher shadow charge, because they cause more congestion to other classes [10]. As each packet is forwarded, the next network meters the remaining downstream shadow price it carries, in order to levy a congestion charge. Because higher priority classes subtract more shadow charges, money flows to resources in proportion to both their demand and the priority of that demand.

Admission control Gibbens and Kelly [9] proposed that path congestion could be used to synthesise connection admission control. Breslau *et al* [1] explored the limits to using this approach on end systems, where, without the ability to enforce self-admission control, guarantees would be weak. Edge gateway solutions [11] solve this problem, but require edge to edge feedback from egress to ingress gateways. However, if the two gateways are owned separately, information asymmetry between them can allow one to act against the interests of the other. With re-feedback the ingress gateway can reliably determine path congestion and decide unilaterally on admission control, conferring with the egress only if it chooses. It can then set up flow policers to bar non-reserved traffic from downstream guaranteed treatment.

Dynamic pricing It will still sometimes be appropriate to apply a market price to the downstream path shadow price (DPSP), but without the 'receiver-pays' constraint. Then, rather than the network degrading the service of packets with higher DPSP, the sender's agent will, but with an elasticity dependent on willingness to pay. The receiver can of course arrange to settle at least part of the sender's charge either directly or through a clearinghouse [2].

4 Routing

Assuming a large majority of corresponding end-points are using re-feedback, then all data except

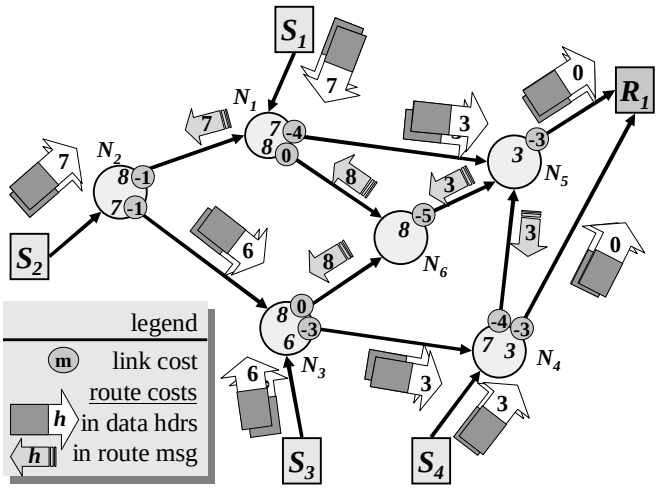

Figure 3: Routing using re-feedback.

the first window of each flow (flagged as uncertain by the source) will carry a reliable and recent characterisation of its remaining downstream path, which every relay it passes through can use. Figure 3 shows this effect, where the large arrows represent data all destined for R_1. $S_1 - S_4$ are *potential* sources of data and in the figure they happen to all be *actual* sources. The squares may be hosts and the circles routers, or at a larger scale, they might all be separately addressable subnetworks. Data is shown carrying a path characterisation metric, for example propagation time. In practice data would carry a vector of metrics and other traffic would be being forwarded to other destinations.

Each egress interface of each router is shown (smaller circles) holding the link cost of its locally connected downstream link, (m, maintained locally (if this were a mobile ad hoc network they would update more regularly). As each router accepts data, it increments the metric by the negative propagation delay m of the logical link (cf. decrementing TTL). Thus all data flowing towards R_1 not flagged as 'uncertain' will carry a metric h in its header representing the remaining delay to R_1.

We are not saying that re-feedback is any use for *creating* routing tables, because that would require a circular dependency. But it allows routing table entries to be maintained continuously *for routes in use*, by overloading header fields that are to some extent already present in IP anyway. In the figure, we show traditional routing messages being beaconed outwards from destination R_1 in order to create routing state. However, where data is flowing towards R_1, it suppresses these routing messages for that link. Instead, the router can sample data it forwards, maintaining an exponentially weighted moving average of each metric.

We have highlighted re-feedback's apparent implications for routing, however, we are yet to work out how best to exploit it in this regard. At the minimum, re-feedback would allow everyone to continuously check the veracity of routing advertisements. Zhu *et al* [16] point out that the conflicting interests of competing networks make it increasingly hard to predict whether and when BGP routing will converge, given asymmetric information about internal network status [5]. They propose a separation between structure and quality monitoring, with probes between trusted nodes to test the latter. Re-feedback would provide inherent route quality monitoring, for *all* parties. More ambitiously, re-feedback could be used to continuously compare a route in use with a back-up route, or to balance multi-path routing load, with suitable hysteresis allowance to avoid route flap.

5 Incentives

Here we aim to create an incentive environment to ensure selfish behaviour leads at least approximately to maximisation of social welfare.[4] Throughout this section, we will again use congestion as our example metric. But the discussion would be identical for other metrics (e.g. TCP also treats time as a shadow price, because a long round trip reduces congestion response).

Two main types of self-interest can be identified:

- Users want to transmit data across the network as fast as possible and pay as little as possible for the privilege. In this respect, there is no distinction between senders and receivers, but we must be wary of potential malice from one to the other;

[4] These mechanisms would be optional if mutual cooperation were the social norm.

- Network operators want to maximise the revenues they extract from the resources they choose to invest in. They compete amongst themselves for the custom of end systems.

We will further divide our discussion between encouraging all parties to *act* responsibly and to pass *information* truthfully: actions and words. Actions being the load users apply, the routing decisions routers make and the capacity they assign. Words, being the metrics passed around the re-feedback loop.

5.1 Actions

We have already described how Kelly showed [12] how to optimised social welfare of a whole network by exploiting the above self-interests. Users were charged the sum of the shadow prices of congestion experienced at resources along the path they were using (by applying a charge to ECN markings emerging at the receiver). It was assumed that users knew the utility they derived as a function of their data rate. User agents would then continuously alter their data rate to maximise the balance between their utility and the congestion charge. Networks would alter the rate of bulk congestion marking at a resource dependent on its total load. With each party acting selfishly and solely on local information, it was shown that the proposed mechanism would lead to behaviour that maximised social welfare, under fairly realistic assumptions.

With re-feedback, we stand on the shoulders of Kelly with regard to proving incentive compatibility, but we fix the *direction* in which shadow prices are applied. With regard to *end-points*, we have already discussed the shortcomings of the 'receiver pays' model and why re-feedback solves them, in the process enabling non-monetary (shadow) penalties on senders (§3). We believe this will be more common than congestion charging as a *retail* tariff.

However, we do believe wholesale and interconnect markets will tend towards congestion pricing, as the result of intensifying competition. For this reason, we will assume without too much loss of generality that congestion pricing is the default *interconnect* tariff to make the following discussion concrete.

A routing incentive has to be created for network A upstream of resources B & C to send traffic to D through whichever of B & C is least costly. For any metric that accumulates along a path, charges can be raised by metering the metric as data crosses each customer-provider interface. Money then flows to each network in proportion to the amount it increments the metric. If the common origin of the metric is at the sender (as with ECN and TTL), money will flow from the receiver *against* the data flow. This is because the metric represents the *upstream* path cost. But then, network D has to pay B or C for the consequences of A's actions. Further D has to pay for the consequences of the routing decisions of all networks upstream of A. Re-feedback fixes this problem, because a common receiver origin makes money flow in the *same* direction as the data. So A pays for the consequences of its own actions, because each point of control and the relevant information needed for that control are co-incident.

5.2 Words

A common misconception is that congestion pricing between network operators is flawed because any network can just fake congestion notification in order to raise excess revenues. Firstly, overstating congestion simply causes demand to reduce unnecessarily (the response of congestion control algorithms), so whether revenues (which depend on the product of price and demand) increase will depend on demand elasticity — usually revenues will reduce. On timescales of minutes in a competitive market, if one operator fakes congestion, another can capture *all* its revenues by offering cheaper bypass routes. So, among networks of equal capacity, the most honest will win. Where there is insufficient competition for a route, excess profits can be made until there is sufficient incentive for other network operators to provision competing capacity. Finally, on time-scales of months, persistent fake congestion reduces customer utilisation of links, which will in the long run also reduce revenues from capacity subscriptions.

Networks that persistently *understate* their costs in routing adverts will draw traffic from competitors, but not have the capacity to serve it, so their neighbours will soon ignore their routing adverts.

Turning now to the honesty of end-points, a sender is discouraged from *overstating* downstream congestion by the incentives we described in §3. That is, if an unnecessarily high shadow price is placed in sent data, penalties will be unnecessarily applied (whether service degradation or congestion charges). Figure 4 shows this effect to the right of the origin, where net value decreases monotonically with overstatement, ρ_c, of the DPSP.

This leaves a clear incentive for the sender to *understate* the downstream path shadow price (DPSP) (net value would continue to rise to the left of the origin in the figure). We solve this by expecting no-one to pay for traffic once its DPSP drops below zero. In other words, a positive DPSP value implies sender pays, but a negative DPSP value implies no-one pays (*not* receiver pays).[5] Then, any network

[5] The introduction of this mathematical asymmetry raises concerns that the model may not be correct. But, it is, in fact, a consequence of the asymmetry of congestion, which does not go negative when a resource is under-utilised.

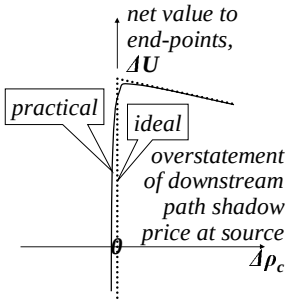

Figure 4: Truth telling incentives

can be certain that traffic with negative DPSP no longer carries any ability to pay for any further congestion it may encounter downstream. Therefore it should be dropped.

The resulting value of understatement is shown dropping away to the left of the origin (figure 4). It can be seen that the sender maximises its net value through honesty where ($\rho_c = 0$). In fact, the position is complicated by continuous variability of path congestion. So although negative DPSP traffic no longer has any capacity to cover its costs, a network will want to make allowances for path variability. A dropper that makes such allowances but still detects malicious understatement of DPSP is described below. An ideal dropper would give a sharp incentive to be absolutely honest, but the best dropper we can currently envisage suffers some false positives and false negatives due to path variability.

A receiver that genuinely wants data to be sent as quickly as possible has incentives aligned with the sender, so honest feedback also returns the maximum net gain. However, if the sender is paying congestion charges (rather than a service penalty), a receiver's incentives are more ambivalent. If the receiver's interests are independent of or counter to the sender's financial interests, the receiver can inflate its feedback to cause the sender to spend excessively. However, a sender will usually only be willing to pay a premium to deliver to known correspondents.

Dropper Given congestion variation during path traversal, let's assume that the probability distribution of DPSP on arrival at the receiver is $P_n(\rho_n)$, centred around zero for a truthful sender. Then the resulting distribution, if the sender were persistently under-declaring the initial DPSP by $\Delta\rho_c$, would be shifted to $P_n(\rho_n - \Delta\rho_c)$.

We propose a dropper[6] at the last hop before the receiver which penalises each packet with a probability dependent on *both* how negative each packet is *and* the exponentially weighted moving average DPSP, μ of recent packets to the same destination. Then no recent history of misbehaviour ensures no drops. But if the average moves negative, the most negative packets will be more likely to be penalised. Packets with the 'uncertain' flag set (see §1) are ignored for the mean, but not for the penalty. If we assume for now just one flow, $\mu \to \Delta\rho_c$.

Figure 5 shows the probability distribution of the DPSP of a cheating packet (before policing), with a conjectured penalty probability function superimposed on a different vertical scale. We want no more negative packets to be penalised than there are positive DPSP packets, so that the distribution of unpenalised packets (without shading) mirrors positive packets. Thus, for $\rho_n < 0$:

$$(1 - p(\rho_n, \mu))P(\rho_n - \Delta\rho_c) = P(\rho_n + \Delta\rho_c). \quad (1)$$

If ρ_n is normally distributed we can find the penalty function in terms of ρ_n, μ, σ, where σ is the standard deviation of the distribution, found by prior measurement of packets known to be honest[7]. Rearranging (1) and substituting from the standard formula for a normal distribution (using base 2):

$$p(\rho_n, \mu) = 1 - 2^{\frac{-2\rho_n\mu}{(\ln 2)\sigma^2}}; \qquad \mu < 0, \rho_n < 0 \quad (2)$$

This penalty function becomes stricter the worse the mean becomes while balancing negative and positive traffic.

Where a malicious flow is hidden in a large aggregate, it will cause a slightly negative mean, leading to some dropping. After Floyd and Fall [7] we cache the flow identifiers of penalised packets. Once any pattern of source or destination identifiers appears that would be unlikely by chance, traffic matching those identifiers is filtered into a second instance of the dropper, which maintains its own mean and may spawn further droppers. Assuming some level of source address spoof prevention, these more targeted droppers should be far more sensitive than the first.

Having found suspect identifiers, a dropper could send hints upstream. Once an upstream dropper had satisfied itself that the recommended focused discard was worthwhile, it could pass the hint further upstream still, thus filtering closer and closer to the problem. These hints could not form a potential DoS attack themselves, as nodes can test the hints.

5.3 Initial window incentives

At the start of each flow, the state of the downstream path is unknown. Currently, a source can start at full rate with no penalty if the path turns

[6] Various penalties short of dropping, e.g. payload truncation, can be imposed given a packet may be wrongly penalised.

[7] For bursty cheating, the deviation could never be less than the original distribution.

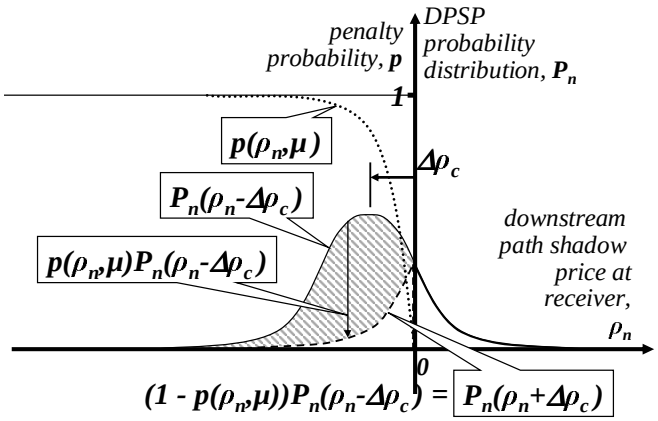

Figure 5: Penalising misbehaviour under uncertainty

out to be clear, or with only as much loss as it causes everyone else otherwise. With re-feedback, all risk is placed on the sender. If it understates initial DPSP, packets may get dropped. If it overstates, it will receive degraded service. This encourages behaviour reminiscent of TCP's slow start, making a network robust even if traffic is dominated by short flows.

This feature makes re-feedback suitable as the first line of defence against denial of service in unicast datagram networks, in two ways. Firstly, as DoS is fundamentally congestion of a resource, it is simply treated as such. Upstream relays focusing load on the victim inherently *pull back* congestion from the problem area, sharing the penalty proportionately across the causes of congestion. Secondly, if an attacker tries to evade paying the proportionate cost of its actions (whether through service degradation or financial penalty), its traffic can be dropped.

This still leaves the system vulnerable to attackers willing to commit resources that match the total effect[8] on all other users, or to those able to harness the resources of others. But our initial ambitions with this work do not extend beyond making the system compatible with the incentives of rational people.

6 Related Work

Clark [4] proposed a decrementing field representing payment as a packet traversed a path. We argue that a field should represent a property of the path, then a price may be *applied* to it. Otherwise a field not bound to a feature of the service can be altered separately from the service. Savage *et al* [14] proposed the ECN nonce as the elegant mechanism for a sender to detect feedback suppression by a receiver. However, it only works if the sender align its interests with the network, against its interest.

[8] That is, costs visible to the communications system, which will sometimes exclude consequential losses.

7 Conclusions & Further Work

We have argued for a re-alignment of the common reference of path characterisation metrics at the destination of any fast feedback loop in unicast packet networks. Downstream information is then located at the same place as the control mechanisms that need it, such as congestion control and routing. Further, downstream path characterisation arrives with each packet, albeit a round trip delayed, giving any relay a view of the remaining path to be traversed by the packet.

We have briefly surveyed some possible ways to use re-feedback, covering routing and four styles of congestion control. Because re-feedback ensures all parties on the path can see downstream path information, it enables a tussle over who controls the network service. It allows senders a view of route costs, and networks a view of downstream congestion. We have also shown how re-feedback can be proofed against the selfish strategies of parties in the feedback loop.

Much remains to be done. A lot hinges on whether we can tune the dropper to catch malicious flows early enough without too many false negatives. Then a formal incentive analysis will be necessary, including considering various collusion models and dynamic attacks. We plan to develop and verify the designs of the congestion control and routing schemes. We also plan to write-up our ideas on how to ease a transition from existing IP feedback to re-feedback, considering environments with more heterogeneous service models and charging plans.

Acknowledgements

Dave Songhurst, Keith Briggs, Nigel Walker, Marc Wenninck, Carla Di Cairano-Gilfedder, Martin Koyabe (BT Research) Jon Crowcroft (Cambridge Uni).

References

[1] Lee Breslau, Edward W. Knightly, Scott Shenker, Ian Stoica, and Hui Zhang. Endpoint admission control: Architectural issues and performance. *Proc. ACM SIGCOMM'00, Computer Communication Review*, 30(4), October 2000.

[2] Bob Briscoe. The direction of value flow in multi-service connectionless networks. In *Proc. International Conference on Telecommunicatons and E-Commerce (ICTEC'99)*, URL: http://www.btexact.com/projects/mware.htm, October 1999.

[3] David Clark, Karen Sollins, John Wroclawski, and Robert Braden. Tussle in cyberspace: Defining tomorrow's Internet. *Proc. ACM SIGCOMM'02, Computer Communication Review*, 32(4), August 2002.

[4] David D. Clark. Combining sender and receiver payments in the Internet. In G. Rosston and D. Waterman, editors, *Interconnection and the Internet*. Lawrence Erlbaum Associates, Mahwah, NJ, URL: http://diffserv.lcs.mit.edu/, October 1996.

[5] Ioanna D. Constantiou Costas A. Courcoubetis. Information asymmetry models in the Internet connectivity market. In *Proc. 4th Internet Economics Workshop*, URL: http://www.m3i.org/papers/ie.pdf, May 2001.

[6] Sally Floyd. TCP and explicit congestion notification. *ACM SIGCOMM Computer Communication Review*, 24(5):10–23, October 1994. (This issue of CCR incorrectly has '1995' on the cover).

[7] Sally Floyd and Kevin Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, August 1999.

[8] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[9] Richard J. Gibbens and Frank P. Kelly. Distributed connection acceptance control for a connectionless network. In *Proc. International Teletraffic Congress (ITC16), Edinburgh*, pages 941–952, URL: http://www.statslab.cam.ac.uk/~frank/dcac.html, 1999.

[10] Richard J. Gibbens and Frank P. Kelly. On packet marking at priority queues. *IEEE Transactions on Automatic Control*, 47(6):1016–1020, June 2002.

[11] Martin Karsten and Jens Schmitt. Admission control based on packet marking and feedback signalling – mechanisms, implementation and experiments. Technical Report TR-KOM-2002-03, TU-Darmstadt, URL: http://www.kom.e-technik.tu-darmstadt.de/publications/abstracts/KS02-5.html, May 2002.

[12] Frank P. Kelly, Aman K. Maulloo, and David K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[13] Andrew Odlyzko. A modest proposal for preventing Internet congestion. Technical report TR 97.35.1, AT&T Research, Florham Park, New Jersey, URL: http://www.research.att.com/~trmaster/TRs/97/97.35/97.35.1.body.ps or URL: http://www.research.att.com/~amo/doc/modest.proposal.pdf, September 1997.

[14] Stefan Savage, Neal Cardwell, David Wetherall, and Tom Anderson. TCP congestion control with a misbehaving receiver. *Computer Communication Review*, 29(5):71–78, October 1999.

[15] Vasilios A. Siris. Resource control for elastic traffic in CDMA networks. In *Proc. ACM International Conference on Mobile Computing and Networks (MobiCom'02)*, URL: http://www.ics.forth.gr/netlab/wireless.html, September 2002. ACM.

[16] Dapeng Zhu, Mark Gritter, and David R. Cheriton. Feedback based routing. *Computer Communication Review*, 33(1):71–76, January 2003.