In Advance Activation of Backup Channels for Real-Time Transmission

Enrique Hernández Orallo and Joan Vila Carbó
Departamento de Informática de Sistemas y Computadores. Universidad Politécnica de Valencia

Abstract

Transmission reliability and availability has to be provided in a QoS framework. Two main approaches have been introduced for guaranteeing Quality of Service: Integrated Services and Differentiated Services. In the Integrated Services (IntServ) approach a real-time channel is established for each individual flow. Differentiated Services (DiffServ), on the other hand, works with traffic aggregates instead of individual flows.

The mechanism for failure detection is often based in timeouts. The latency of failure detection is a key aspect in providing an uninterrupted service (availability). Packet delays and timeouts can be estimated using “Network Calculus” which is an analytical framework for estimating delays, as a function of the bandwidth reservation [8].

We introduce a new scheme using backup channels. Two ideas are the basis of this scheme: first, the end-to-end probability density function (PDF) has a long tail (a great percentage of packet delays are very far from its deadline). Second, the longer is the failure latency time, the lower the delay requirements of the backup channel are (and this implies higher network reservations).

According to this, the paper introduces a scheme for fault detection called Proactive Backup Channel (PBC) whose goal is to reduce failure latency (and, thus, resource reservations). PBC reduces failure latency by “suspecting” a failure before it occurs. This technique can be considered a trade off between failure detection latency and the utilisation of the backup channel.

In this paper we focus our experiments in the Differentiated Services Network. The solution presented is a generalisation of a scheme introduced by the authors for Video transmission in Integrated Services [9][10].

2. Failure Detection Scheme

This section presents the basis for a new failure detection scheme. There are two main approaches to provide fault-tolerance for real-time channels:

- **Multiple copy** (MC): several redundant real-time channels are set-up through disjoint routes and packets are sent simultaneously through all channels. There is no need for a failure detection mechanism: the scheme works as long as the
receiver gets at least one of the replicated packets. This is a costly technique (all the channels are permanently used) but it provides uninterrupted service in spite of failed routers.

- **Reactive Backup channels** (RBC): a primary channel and a backup channel are required, but the backup channel is only set-up upon a failure of the primary channel. That requires a failure detection mechanism. Since the backup channel is not used until a failure occurs, this technique saves unnecessary bandwidth utilisation, but an additional delay is introduced for setting up the backup channel. Besides, it may occur that there exist not enough resources when trying to set-up the backup channel.

Reserving resources in advance for the backup channel can solve some drawbacks of the second approach: that would improve the set-up time and avoid the lack of enough resources. Resources for the backup channel are reserved but not used in absence of failures, so they can be used to transmit non critical traffic.

This paper introduces Proactive Backup Channels (PBC), a new failure detection aimed to achieve an efficient channel resource reservation by reducing the failure detection latency. It is based on “suspecting” a failure when the packet delay through the primary channel is close to the maximum guaranteed delay. Whenever a failure is suspected, the backup channel is activated although the primary channel would not be discarded. This yields unnecessary backup channels activations due to inaccurate failure detection (false failures), but it has the benefit that the delay requirements for the backup channel are not so restrictive. That allows reducing the required resources for the primary and backup channel.

The efficiency of the PBC scheme is based in the following two typical network behaviours:

- **End-to-end delay distribution**: Packet end-to-end delays follow the typical distribution of Figure (1) so a given activation delay that corresponds to 99% of the packets can be easily derived from this curve. If a packet time is higher than the activation delay the client start to set-up backup channel.

- **Network resource utilisation** decreases when the required end-to-end delay is increased (figure 2).

One goal of this paper is, given a packet maximal end-to-end delay \(d_{\text{total}}\), for a primary channel, which are the necessary resources (bandwidth) to be allocated for the primary and the secondary channel. This allocation strongly depends on the allowed delay for each channel. Let \(d_f\) be the failure detection time and \(d_s, d_B\) the end-to-end delay assigned to the primary and secondary channel. For the MC approach as there is no failure detection the channels delays are simply \(d_{\text{total}} = d_s + d_B\). In the RBC approach \(d_f = d_A\) so the maximum delay experienced by the first packet retransmitted through the backup channel upon a failure can be expressed as \(d_{\text{total}} = d_A + d_f + d_B\). In the PBC approach \(d_f < d_A\) so the end-to-delay can be now expressed as \(d_{\text{total}} = \max(d_A, d_f) + d_B\). This way, the upper bound for \(d_A\) could be \(d_{\text{total}}\), while \(d_B\) could be chosen as \(d_{\text{total}} - d_f - d_s\). The fact that \(d_A\) is \(d_{\text{total}}\) implies an important reduction in resource reservation for the primary channel. Then, the total resource reservation will depend on the selection of \(d_f\). This value must be carefully selected depending on the end-to-end delay distribution. A lower value of \(d_f\) implies a higher \(d_B\) value; that would lead to a reduction in the bandwidth requirements for the backup channel, but would produce false failures. For example, for the distribution of figure 1, with a maximal end-to-end of 400 ms and setting \(d_f = d_{\text{total}}/2 = 200\) ms this would imply that 0.1% of the packets would produce a false failure activation. As will be shown in section 4, the set-up time for the backup channel \((d_A)\) is insignificant when is compared with the total delay \((d_{\text{total}})\). Therefore, to simplify, \(d_f\) is assumed to be 0 in the rest of the paper.

![](image1.png)

**Figure 1: End-to-end delay distribution**

![](image2.png)

**Figure 2: Typical bandwidth reservation**

As mentioned in the former paragraphs, the first factor that affects the performance is the packet delay distribution. For a given failure detection time \(d_f\) the false failures rate (\(\Phi\)) can be obtained as the percentage of packets whose delay is above \(d_f\). In the example of figure 1, for \(d_f=200\) ms the \(\Phi\) value is approximately 0.1%. The second factor is the relation between deadline and network reservation. This depends clearly on the network reservation scheme and will be further evaluated for Differentiated Services networks. Nevertheless, a general expression can be introduced here.

Let \(R\) be a function that gives the bandwidth reservation for a given delay.
This function is decreasing, so if \( d > d' \) then \( R(d) < R(d') \). Then, in the MC scheme the total reservation is \( R(d_{1}) + R(d_{0}) = 2 \times R(d_{total}) \). In the RBC scheme the total reservation is \( R(d_{1}) = R(d) \).

For PBC the effect of false failures (the cost of activating the backup channel due to false failures) has to be taken into account. Given a false failure rate \( \Phi \), the equivalent wasted resources for false failure will be \( \Phi \times R(d_{b}) \). Therefore, the total reservation for PBC will be \( R(d_{total}) + \Phi \times R(d_{b}) \). Then, PBC scheme will be more efficient than MC if \( \Phi \times R(d_{b}) < R(d_{total}) \), and is more efficient than RBC if \( R(d_{total}) + \Phi \times R(d_{b}) < R(d) \). It is easy to see that the lesser is the \( \Phi \), the greater is the efficiency. Summing up, Figure (3) gives a graphical description of the delay allocation and resource reservation for the three schemes.

In summary, the proposed scheme is a trade-off to allow inaccurate failure detection in order to minimise latency optimising, thus, resource reservations. The only situation where the inaccuracy of the failure detector could be considered non-acceptable would be when the sender has periods of relatively long inactivity. This would imply either to inject heartbeats or to modify the traffic to ensure that the connection is not used for some (relatively) long time.

![Figure 3: Temporal activity of channels](image)

### 3. Differentiated Services

The goal of the **DiffServ** architecture is to provide differentiated classes of service for Internet traffic, to support various types of applications, and specific business requirements [12]. Packets are classified and marked to receive a particular per-hop forwarding behaviour (PHB) on nodes along their path. Network resources are (pre-)allocated to traffic aggregates by service provisioning policies which govern how traffic is marked and conditioned upon entry to a differentiated services-capable network, and how that traffic is forwarded within that network.

In order to provide deterministic guarantees and implement a reliable backup system we used the EF (Expedited Forwarding) class of service [13].

#### 3.1. Analytical end-to-end packet delay

EF provides a means to quantify the maximum delay that a packet may experience under bounded operating conditions. As defined in [13] the delay bound of any packet departing from a node is:

\[
D = \frac{B}{R} + E_{p}, \quad r \leq R
\]

where \( R \) is the configured EF service rate on the output interface, and the total traffic destined to the output interface is bounded by a token bucket of rate \( r \leq R \) and depth \( b \). \( E_{p} \) is an error term for the treatment of individual EF packets. The value \( E_{p} \) depends on the particular implementation of a node (in [14] it is detailed how to calculate this value for several scheduling disciplines).

The multi-hop worst case packet delay in a **DiffServ** network \( D \) can be obtained from equation (1) as the sum of the local delays in each hop \( D_{p,h} \):

\[
D = \sum_{h=0}^{H} D_{p,h}
\]

If \( R \) is equal for all nodes and \( b \) is the worst case input burstiness across all nodes of the path then:

\[
D = \frac{H \times B}{R} + E_{tot}, \quad r \leq R
\]

where \( E_{tot} \) is defined as the sum of the \( E_{p} \) parameters defining each node of the aggregate path.

As a result, solving out for \( R \) we can obtain the reservation function:

\[
R = \frac{H \times B}{D - E_{tot}}, \quad r \leq R
\]

Therefore, we have the first condition for applying the PBC scheme. It provides the necessary bandwidth reservation to accomplish a given delay. The second condition (the delay distributions) will be evaluated in the following sections.

It is worth noting that the EF channel parameters \( (B \) and \( R \) are static, that is, they are selected when the network is dimensioned. Therefore, an admission control is needed in the ingress nodes to admit (or reject) channels in order to conform the traffic aggregate to the leaky bucket parameters.

#### 3.2. PBC in Differentiated Services

The implementation of PBC in **DiffServ** is similar to the Integrated Services scheme except that the channel is a traffic aggregate channel. The solution is based in creating a primary and backup channel in the **DiffServ** cloud for all the traffic that has the same origin and destination (see figure 4). The primary channel will be configured with a reservation \( R \) and buffer \( B \). This means that a packet that traverses the **DiffServ** network from an **Ingress** node to an **Egress** node will have a bounded delay \( d_{total} \). Then, applying the PBC scheme is easy: if a packet...
is delayed more than a given \( d_f \) time then the backup channel is activated. The backup channel will be configured with \( R \) and \( B \) parameters to provide a \( d_b=d_{\text{total}}-d_f \) bound. The main difference with the integrated service approach is that in DiffServ, the PBC scheme works with traffic aggregates and not the primary and backup channel are statically configured (when the network is dimensioned).

**Figure 4: DiffServ PBC implementation**

The Egress node has the responsibility of detecting a fail (a packet is delayed more than \( d_f \)). When a fail is detected the entire traffic aggregate switches to the backup channel.

An improvement over the presented scheme consists of providing a primary and a backup channel for each path. As stated in the appendix of [13], it is quite acceptable for a DiffServ network to provide multiple instances of EF class of service. Then, each DiffServ node would be configured with two EF levels (EF1 and EF2). The EF1 is used to transmit the traffic aggregate of the primary channel. When a fail is detected in the path, the aggregate is transmitted using a backup channel with the EF2 level. This would imply only a degradation of low priority traffic transmitted through the backup channel (AF and best-effort classes). If there were no failures, this EF2 class would not be used, so it would not affect lower priority traffic.

This is a very interesting property for Internet Backbones providers. This approach would provide a failure backup scheme with no reservation in case of absence of failures and little degradation of low priority traffic when a failure is detected. Therefore, the backbone provider could design its network with disjoint paths that are efficiently used.

### 3.3. Experimental evaluation

This section presents the experimental evaluation of the efficiency of the PBC scheme in a DiffServ network. This evaluation has been done with a test program called RTNOU (Real-Time Network Optimisations Utilities). This program can be freely downloaded from Internet from the following web site: http://www.disca.upv.es/enheror/RTNOU.html

The evaluation requires defining a traffic aggregate. We used the MAWI traffic traces [15] due to their high resolution (necessary for simulation) and their relative high mean bandwidth. Concretely, we took a 24-hour trace of May 14, 1999 from a US-Japan link. This is typical internet traffic as can be shown in figure 5.

**Figure 5: MAWI traffic (using one-minute sample period)**

These traces were in tcpdump raw format (near 9 Gbytes of traces), so we distilled them to obtain a simple file that contains the number of bytes transmitted for a given period in each line. The resulting traffic trace has 8,639,919 frames and a mean rate of 6,617,785 bps and a peak rate of 33,960,800bps. The period used is 10ms (100 samples per second).

Using this traffic aggregate we can compare the network resource reservations for the three schemes following the expression of figure (2). The network used is depicted in Figure (6) using a typical DiffServ Strict Non-preemptive Priority Queue (PQ) scheduler in the nodes. For the PQ policy, the latency term \( E_p \) is \( MTU/C \) where \( MTU \) is the maximum packet size and \( C \) is the speed of the output link [14]. Then, for the evaluation network, \( E_{\text{tot}} = 0.00109s \) and \( H=4 \). The channel parameters are obtained using an optimisation method that is equivalent to the IntServ method [16]. In short, given a deadline and a traffic aggregate, parameters \( B \) and \( R \) are selected in order to obtain the minimal network bandwidth reservation. For example, for a delay of 0.01s the channel parameters are \( R=27,775,738b/s \) and \( B=61,850bits \).

Table (1) shows the results of the total bandwidth reservation in a node for two different end-to-end delays: 0.02s and 0.2s. Without loss of applicability, \( d_f \) will be \( d_{\text{total}}/2 \) and the false failures rate (\( \Phi \)) is assumed to be 0.001. The Reduction column shows the reduction of resources of the PBC scheme: \( \text{Reduction}=100\times(1-R_{\text{PBC}}/R_{\text{MC,RBC}}) \). The new scheme PBC achieves a remarkable reduction versus the other schemes. The results show that for both delays (0.02s and 0.2s) the bandwidth reduction is high. Comparing the PBC with the MC scheme the bandwidth reduction is practically constant and near to 50%. Regarding the PBC versus RBC schemes, the reduction is about 17%. The resource gain depends mainly of the difference between the bandwidth reservation obtained for \( d \) and \( d/2 \).
The efficiency of the PBC depends on the delay distribution of the aggregated traffic. Therefore, the following experiment obtains the traffic distribution for a deadline of 0.02s. In the simulation scenario the traffic aggregate is introduced in the network at the Ingress node. The load in the network is introduced by creating 5 AF channels in each node with a load index that ranges from 0% (no load) to 100% (full load). The optimal channel parameters for a 0.02s deadline are: bandwidth reservation R = 23,060,565 bps and B = 109,002 bits.

Figure (7) represents the density functions for packet arrivals in the simulations for different load indexes of the aggregated traffic: all packets arrive much sooner than their nominal deadlines and the more loaded the network is, the more the packets are delayed. For full load the maximum packet delay is about 0.0017s far beyond the nominal deadline (0.02s).

These results show that there are practically no false failures, so the PBC scheme is very efficient. One question that arises is why the packet delays are so much lower than their deadlines. There are two main reasons: first, is the bursty characteristics of the traffic and second, is that traffic characterisation and delay equations (as equation (1)) makes a very coarse approximation of the traffic dynamics so delay bound are very pessimistic.

Summing up, these experiments show that the PBC scheme can reduce considerably the bandwidth reservation for providing backup channel in DiffServ networks.

4. PBC Implementation

The Proactive Backup Scheme uses an end-to-end failure detection scheme. The receiver node has the responsibility to detect if the channel fails and to begin the activation of the backup channel. The source node injects a heartbeat into the channel when the sender is idle. On the other hand, the network must provide mechanisms for establishing disjoint paths.

Therefore, the PBC Scheme can be easily implemented as a part of a transport protocol on top of a network protocol. The FSM (Failure Suspect Module) module will provide the mechanisms to select the primary and backup channels, activate the backup channel when a packet delays more than \( d_f \) and inject heartbeats in the network when there is not traffic. However, in the experiments done with MPEG video traffic aggregate, it has been proven that the injection of heartbeat packets is not necessary because there is always traffic to send. The FSM can be placed in the ingress/egress nodes of a DiffServ network.

The most important issue of this new scheme is failure detection. In PBC, the activation of the backup channel starts when a packet delay is greater than \( d_f \). But, how the receiver knows if a packet is delayed more than \( d_f \)? In the proposed solution the sender includes the estimated departure time of the following packet (\( t_{prox} \)) in every packet. When this packet reaches the receiver, the failure condition will be if the following packet arrives later than \( t_{prox} + d_f \). This mechanism implies that the sender and receiver must be synchronised, that is, must have the same time base.

Figure (8) shows a sample of how this method works. The sender sends a first packet with time \( t_1 \) and the time of the following packet \( t_{prox} = t_2 \). When this packet reaches the receiver, it knows that the next packet will arrive before \( t_3 + d_f \). For example, if packet 3 is lost due to a...
primary channel failure, then the FSM assumes a failure when time is $t_f + \tau_f$ and starts the activation of the backup channel sending a NACK message to the sender (using the backup channel). When the NACK message reaches the sender, it starts transmitting through the backup channel, keeping the transmission through the primary (the backup channel resources are always reserved, so the backup channel is ready to transmit). If the fail is confirmed (the receiver sends an NACK-C message) the sender will stop sending by the primary channel. If the fail is not confirmed, the receiver will send an ABORT message to the sender to stop the transmission by the backup channel. Therefore, in the case of a false failure this would imply the transmission of only 2 or 3 packets.

**Figure 8: Failure detection and backup activation sequence**

In order this scheme to work correctly, the network must provide a bounded and short set-up delay $d_s$. The path for this NACK message will be the backup path (it assumed to be OK). We stated in section 2 that the setup time for the backup channel ($d_s$) is negligible compared to the total delay ($d_{total}$). In the PBC implementation proposed in this section the setup time $d_s$ is the time needed for a high priority message to reach the sender. In the simulations this time is very low (it ranges from 15μs for the ATM like network to 187μs for IP networks [11]) so the assumption for $d'_s=0$ is acceptable.

5. Conclusions

This paper introduces the Proactive Backup Channels (PBC) scheme in order to reduce the resource reservation for highly available real-time channels. The idea behind this scheme is that packet delays are far behind their analytical bounds, so it is not necessary to wait for the primary channel deadline in order to activate the secondary channel. This is known as a proactive activation versus the traditional reactive activation.

The PBC scheme has been compared with the Multiple Copy (MC) and Reactive Backup Channel (RBC) approaches in Differentiated Services Networks. The experiments show that the new scheme is very efficient and can provide savings near 17% of network resources. Another advantage of this scheme is its easy implementation. The mechanism can be implemented as part of the transport level so no modification is needed at the network level. This new scheme is a generic solution so it can be applied to other type of networks. For example, the implementation in Integrated Services networks showed that it can reduce bandwidth requirements in near 50% over traditional schemes [10].

References