Multiplexing Real-Time Channels

Spiridon Damaskos

Gesellschaft fur Mathematik und Datenverarbeitung mbH
Berlin, West Germany

Dinesh C.Verma

Computer Science Division
EECS Department
University of California
and International Computer Science Institute
Berkeley, California

Abstract

A real-time channel is a simplex connection between two nodes characterized by parameters representing the performance requirements of the client. Such a connection may be established through the scheme described in [Fer89a]. In this paper, we study the feasibility of multiplexing real-time channels on a lower-layer real-time channel. Sufficient conditions for multiplexing channels are obtained as an extension of the establishment algorithm.

The extension is based on two observations: (1) a real-time channel can be looked upon as a network with bounded delays connecting the multiplexing point (a virtual source) to the demultiplexing point (a virtual destination); and the parameters of the physical channel can be used to define the service time at the virtual source and sink. Multiplexing is nothing but channel establishment over this network. By a judicious definition of the parameter specifying service times, it is possible to make multiplexing decisions at the multiplexing point (source) without consulting the destination, which is merely informed about the new multiplexed channel.

This research has been supported in part by the Defense Advanced Research Projects Agency (DoD), ARPA Order No. 4871, monitored by Naval Electronic Systems Command under Contract No. N00039-84-C-0089, the University of California with a MICRO Program grant, Cray Research, Inc., Hitachi, Ltd., IBM Corporation, Ing. C. Olivetti & C., S.p.A., International Computer Science Institute and Gesellschaft fur Mathematik und Datenverarbeitung mbH. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing official policies, either expressed or implied, of any of the sponsoring organizations or of the U.S. or West Germany Government.

1. Introduction

Real time computer communication, i.e. computer communication with guaranteed performance, is expected to become a necessary feature of future networks. An abstraction which provides such guarantees is the real-time channel[Fer89a] or the parametrized message channel[And88]. The channel serves as a common interface between successive layers of a communication protocol model, and needs to be supported by the lowest layer (i.e. the network layer).

A network real-time channel, unless it is best-effort, requires resources to be reserved at intermediate hosts, source and the destination. We therefore need a scheme which minimizes the number of channels to be established. One way to minimize this number, while providing a separate channel abstraction to each application or client, is to multiplex many *virtual* channels onto a *physical* channel, by which we mean a network real-time channel. We describe a scheme to provide such multiplexing for physical channels established through the algorithm in [Fer89a].

There are many instances where such multiplexing can be advantageous. In multimedia communication, the same application may need to handle different kinds of data. This may require a number of different real-time channels to be set up. If we want to establish either all or none of these channels, we can request a physical channel which can handle all of this data and multiplex many virtual channels on it. In another instance we can have a voice and a video channel with some dependency among them. One such requirement can be that they follow the same route. Multiplexing them on top of a common channel provides an easy solution.

In another possible situation, we can imagine a joint paper being written by two people, one at Berlin, West Germany and the other at Berkeley, USA. These two individuals would like to open several windows and establish real-time channels between corresponding windows. If we can multiplex all the channels onto a single real-time channel, the phone bill can be reduced appreciably.

Let us call a real-time channel provided by the network a physical channel and any channel multiplexed over it a virtual channel. A scheme for managing these channels can be seen in Figure 1. The physical channel manager is responsible for accepting and rejecting requests for a physical real-time channel. His only local client is the virtual channel manager (physical channel managers from other nodes in the network are also his clients). An application desiring a set of real-time channels to another application across the net communicates his requirements to the virtual channel manager. The virtual channel manager would examine all the existing physical channels to the requested destination and determine a suitable physical channel. It would then notify the virtual channel manager at the destination of the existence of the new virtual channel and return a virtual channel descriptor to the application. The application can use this descriptor for data transfer.

In some cases, the virtual channel manager may not be able to find a suitable physical channel. Then the virtual channel manager would request one or more physical channels from the physical channel manager. The parameters of the physical channel(s) have to be determined so that they will be able to satisfy the virtual channel requirements.

We can say that all the physical channel managers together form the *physical channel provider*. All virtual channel managers constitute the *virtual channel provider*. The virtual channel manager is also a client of the *physical channel manager* while the applications that talk to the virtual channel managers are *virtual channel clients*.

The virtual channel manager is called upon to do two different tasks, to check for virtual channel establishment and to combine the parameters of virtual channels. We therefore need a virtual channel establishment scheme and a calculus of parameters. The virtual channel establishment schemes will be a set of tests which can be run by the virtual channel provider to determine if it is safe to multiplex a set of virtual channels over a given physical channel. The calculus of parameters is a set of formulas which allows the virtual channel provider to determine the parameters required of a new physical channel, which can accommodate the traffic on all the virtual channels.

Section 2 describes the physical channel establishment scheme in brief. We describe the virtual channel establishment problem and a solution in Section 3. Section 4 attempts to develop a limited calculus of parameters, limited in the sense that we consider only one characterization of the traffic on a channel. We describe the virtual and physical channel operators in the next section. Finally we mention our conclusions and some areas of further work.

2. The Physical Channel Establishment Scheme

The scheme described in [Fer89a] is in the context of contemporary wide area networks, although it is applicable to other networks as well. The scheme assumes the following descriptions of the traffic on the channel:

- the minimum packet inter-arrival time on the channel $x_{\min}$,
- the minimum value x_{ave} of the average packet inter-arrival time over an interval of duration I or more, and
- the maximum service time t in a node for the channel's packets.

A channel has guaranteed end to end delay bounds. This guarantee may be of three kinds.

- Deterministic: the bound D is an absolute one. All packets on the channel must get a delay of D or less.
- Statistical: the bound is expressed with the delay D and the probability Z that a packet's delay is less than D.
- Best-effort: a delay bound is specified, but no guarantees are made.

The end to end delay bound D is divided into the local delay bounds d^i at each intermediate node i. For a statistical channel, Z is broken up into components z^i for each intermediate node. As the establishment request moves from the source to the destination, each node on the establishment path verifies that acceptance of the new channel is consistent with the guarantees of any existing channels. If so, a suggested value of the local delay bound for this channel is included in the establishment request. The destination does the final allocation of the local delay bounds, it may increase the local delay bounds for the intermediate nodes but can not decrease them. These local delay bounds are committed on the return trip of the establishment message.

Three tests are made by the physical channel manager at each intermediate node during the forward establishment request.

The deterministic test verifies that the guarantees to existing deterministic channels are not violated. It is done by verifying that

$$\sum_i t_i / x_{min,i} \leq 1 , \tag{1}$$

where i ranges over all the deterministic channels and one statistical or best effort channel (the one with the maximum value of $t/x_{\min}$) at the intermediate node. The condition ensures that even when all the deterministic channels are sending packets at their maximum possible rate, the node is not over-booked (saturated).

The statistical test calculates the probability that a deadline can be missed because the node is over-booked, i.e. condition (1) may not be satisfied when we consider all deterministic and statistical channels. It verifies that the probability of node saturation is acceptable to each established statistical channel.

$$Pr\left(\sum_i t_i / x_{min,i} \leq 1 \right) \geq \max_j z_j , \tag{2}$$

where j covers all statistical channels and i ranges over all deterministic and statistical channels.[1]

---

[1] The reader may have noted that in forms (1) and (2), we assume that the time to insert a packet in the queue at a node is small compared to the service time of the packet. If this assumption is not valid, and there is only one server for both inserting the packets in the queue and servicing the packet, the equations will need to be modified. In such a situa-

The delay bound test checks that the node can handle the different deadlines of the packets during the operation of the channel. A set of channels is selected for this test. For establishing a deterministic channel, we must select all the deterministic channels. We make the pessimistic assumption that all channels miss their deadlines whenever (1) is not satisfied. To ensure that the performance guarantees for established statistical channels are met, we should not miss a deadline whenever condition (1) is satisfied. So we must consider the set of all deterministic and statistical channels which satisfies condition (1), and is the worst possible combination i.e. it is very likely to cause a deadline to be missed. Without loss of generality, assume that the channels in this set are $1 .. K$ with $d_1 \le d_2 .. \le d_K$, then, for all channels j such that the local delay bound of j is smaller than the sum of all the service times of all the channels at this node, we need that:

$$d_j \ \ge \ \sum_{i=1}^{j} t_i + \max_{j<l\le K} t_l \ . \tag{3}$$

The best effort channels, in principle, require no tests to be made. There is no resource reservation for them. However, if these packets are to have finite delays (i.e. to prevent starvation of best-effort packets), the physical channel manager may use this test:

$$\sum_i t_i / x_{ave,i} \le 1 \ , \tag{4}$$

where the sum is taken over all the channels at the node, deterministic, statistical and best-effort.

The physical channel operators in each node implement a version of multiple queue earliest due date (EDD) scheduling. In addition they exerts flow control to insure that the channels keep their promises about $x_{\min}$ and x_{ave}.

For further details on the establishment algorithm, the reader is referred to [Fer89a].

3. The Virtual Channel Establishment Scheme

We assume that we have an established physical channel with certain parameters. Zero or more virtual channels are already multiplexed on this channel. The feasibility problem is determining if a new virtual channel request can be multiplexed on the physical channel.

3.1. Problem Statement

Suppose the parameters [2] of the established physical channel are $x^p_{\min}$, x^p_{ave}, D^p, Z^p, I^p and t^p. Let there be n virtual channels, including the ones already multiplexed on the physical channel as well as the new requests. Let their parameters be $x^v_{min,i}$, $x^v_{ave,i}$, D^v_i, Z^v_i, I^v_i and t^v_i. Let us refer to the set of all virtual channels as V. Let E be the set of deterministic, S the set of statistical, and B the set of best-effort virtual channels.

The service time of a packet consists of various components, the time required for mapping virtual channels to physical channels, time spent in flow control (checking that channel clients keep their promises), queue manipulation operations, and transmission. We assume that t^v and t^p refer to the total time required for all of the above operations. This simplifies the tests for channel multiplexing. (see Figure 2)

In order to establish the new virtual channels over the physical channel, we must observe two rules.

(R1). The acceptance of the new virtual channel must be consistent with the load characterization of the physical channel i.e. the physical channel client (and the virtual channel provider) should be able to keep his promises about $x^p_{\min}$ etc. which he had made to the physical channel

---

tion, even best-effort packets may delay deterministic (or statistical) packets by arriving too rapidly. The effect can be taken into account by adding a factor of $t_{ins} / x_{\min}$ for every channel that is not considered in the original equation (1) or (2), where t_{ins} is the time for inserting a packet in the queue.

[2] A list of all the symbols used in the text and their meanings is given at the end of the paper for easy reference.

provider.

- No virtual channel can have a minimum or average inter-arrival time smaller than that of the physical channel:

$$\min_{i \varepsilon V} x^{v}_{min,i} \geq x^{p}_{\min} \ . \tag{C1}$$

$$\min_{i \varepsilon V} x^{v}_{ave,i} \geq x^{p}_{ave} \ . \tag{C2}$$

- The average inter-arrival times are defined over intervals. Therefore,

$$\max_{i \varepsilon V} I^{v}_{i} \leq I^{p} \ . \tag{C3}$$

- The virtual channel manager must ensure that the service time of any virtual channel must not exceed that of the physical channel.

$$\max_{i \varepsilon V} t^{v}_{i} \leq t^{p} \tag{C4}$$

(R2). The physical channel performance parameters must be *"good"* enough to satisfy the virtual channel performance parameters. This implies that

- a deterministic virtual channel can only be multiplexed over a deterministic physical channel. Similarly a statistical virtual channel can only be multiplexed on top of a deterministic or a statistical physical channel. Although it is feasible to multiplex best-effort virtual channels over any type of physical channel, it is better to multiplex them on top of a best-effort channel only. Otherwise, it is a waste of the resources reserved for the physical channel.
- The end to end delay bound of the virtual channel must be more than that of the physical channel:

$$\min_{i \varepsilon V} D^{v}_{i} \geq D^{p} \ . \tag{C5}$$

- Since the two parameters of D and Z determine a bound for only one point on a channel's delay probability distribution, we are unable to make any statements for a delay lower than that of the physical channel, or for a probability bound greater than that of the physical channel. For statistical channels, we must enforce the condition:

$$\max_{i \varepsilon V} Z^{v}_{i} \leq Z^{p} \ . \tag{C6}$$

Conditions (C1) and (C2) are not sufficient to ensure that the physical channel's load characterization is not violated. For example, two virtual channels with an average inter-arrival time of 30 time units each can not be multiplexed over a physical channel with x^{p}_{ave} of 20. The resultant average inter-arrival time of all the virtual channels multiplexed on a physical channel must be greater than that of the physical channel itself. Counting the number of arrivals in a suitably long interval, we get:

$$\sum_{i=1}^{n} 1/x^{v}_{ave,i} \leq 1/x^{p}_{ave} \ . \tag{C7}$$

We would like to derive a similar relationship between $x^{v}_{\min}$ and $x^{p}_{\min}$. However, if we assume independence among the virtual channels, there is no way to prevent simultaneous arrivals from two different virtual channels. On combining two or more virtual channels, the only lower bound on inter-arrival times of the resultant seems to be zero, and no channel with a $x_{\min}$ of zero can be established.

The virtual channel operator might find two or more packets to be transmitted within a very small time. But sending them one after another will violate the bound on $x^{p}_{\min}$. In such a case, there is no certainty that the physical channel operator can honor the delay bound. How can the virtual channel operator deal with this problem? The only option is to hold such packets and send them out at intervals of $x^p_{\min}$. The same effect can be obtained by increasing the deadlines of the packets as if they had arrived at a later time. The scheme can be seen in Figure 3.

3.2. The solution

Assume for the moment that the physical channel is deterministic. It can be looked upon as a network with a bounded delay (see Figure 4). Then, we have a virtual path for each virtual channel consisting of three components:

the virtual source representing the multiplexing point,
the physical channel as a network with a bounded delay, and
the virtual destination representing the demultiplexing point.

Thus the problem of multiplexing virtual channels on a deterministic physical channel can be reduced to establishment problem on the above network. Channel establishment on such a network can be done by the scheme in [Fer89a].

In the case where the physical channel is statistical with a parameter Z^p, condition (C6) implies that the only channels to be multiplexed over it have Z^v smaller than Z^p. In this case the channel to be established on the above network can be assumed to have a delay bound of $D^v - D^p$ with a probability Z^v/Z^p. Let us refer to these delay bounds as the effective delay bounds i.e. D^e and Z^e.

$$D^e = D^v - D^p$$

$$Z^e = Z^v / Z^p$$

To complete the reduction to channel establishment, we must also examine the characterization of the traffic on the network. The average and peak inter-arrival times are obviously the ones specified by the virtual channel client. For the service time, we must consider the time spent in multiplexing operation as well.

Mapping of the virtual channel to the physical channel is an operation that must be done by the virtual channel operator. The transmission of a packet, of course, has to be done by the physical channel operator. All other operations like flow control or queue manipulations can be done by either the physical channel operator or the virtual channel operator. Assume that an upper bound t_m is known for the total time required by the virtual channel operator to map virtual channels to physical channels. Depending on the implementation, the virtual channel operator may do other operations. Let the total time required for all these operations be t_o.

The maximum rate at which the virtual channel operator can push packets to the physical channel operator is determined by t_o. If this is greater than $x^p_{\min}$, there is no danger of ever violating the minimum inter-arrival time promised to the physical channel. If however, this time is less, the virtual channel operator may need to wait for some time before giving the next packet to the physical channel operator. The effective service time of a packet at the multiplexing point is the maximum of these two values i.e.

$$t_s = \max\,(\, t_o \,,\, x^p_{\min} \,)\,.$$

If there are j packets in front, a packet will have a queuing delay t_q of

$$t_q = j t_s\ .$$

At the destination of the virtual channel, the virtual channel operator needs to *unmap* the physical channel packet into the correct virtual channel, and perform any related operations. Let us refer to this cost as t_{dst}. t_m and t_{dst} can be taken as service times at the virtual source and the virtual destination. At the destination, the processing of the packet by the physical channel operator and the unmapping operation by the virtual channel occur consecutively, most probably without any wait in between the two steps. So we can include this service times with the physical channel's service time. This is justified if we use a scheme where the virtual and physical channel operators cooperate (see section 5).

At the sending side, there are two distinct phases in processing, the processing by the virtual channel operator and the processing by the physical channel operator, the two steps separated by a wait in the queue. It is advantageous to include the service time of the virtual channel operator t_o in the physical channel time t^p at the source as well. A virtual channel packet may have to wait because a packet from another virtual channel is being served. This virtual channel may have been multiplexed on another physical channel. If we include t_o as part of t^p, then we are assured that the interference from packets from virtual channels multiplexed on a different physical channel has been accounted for (during the physical channel establishment test). The multiplexing tests, then, can be restricted to the virtual channels multiplexed on the same physical channel.

The relation between t_m, t_o, t^p and t^v can be seen in Figure 2. With the interpretation of these times mentioned in the previous paragraph, we can assume that the service time at the virtual destination is zero for the purpose of virtual channel establishment. The service time at the virtual source should be taken as t_s defined above.[3]

This scheme has the advantage that the virtual channel manager at the source need not wait for a round trip delay before knowing whether multiplexing is feasible on a given physical channel. If there are many physical channels which may be used for a new virtual channel request, the source knows which one to use without consulting the destination. If none of the channels meet the criterion, he may also request a new physical channel to be established (see section 4). For every unsuitable physical channel that may be tried, we save the time for a round trip.

Of course, the virtual channel manager at the source needs to notify his peer at the destination of this new virtual channel. The virtual channel client can be given a response only after an affirmative response from the destination has been received.

For no node saturation to occur, we must make sure that the total amount of work received by the multiplexing point never exceeds the time it has to process the packets. This corresponds to the deterministic test and can be written as:

$$\sum_i t_s / x^v_{min,i} \leq 1 \, , \tag{C8}$$

where the sum is taken over all the deterministic virtual channels multiplexed over the physical channel. We may call this the *virtual deterministic test*.

An analogous *virtual statistical test* has to be performed when multiplexing statistical channels over a deterministic ($Z^p=1$) or a statistical physical channel ($Z^p<1$).

$$Pr\left(\sum_{i \varepsilon E \bigcup S} t_{s,i} / x^v_{min,i} \leq 1 \right) \geq \max_{j \varepsilon S} Z^e_j \, , \tag{C9}$$

The *delay bound test* has to be performed as in the original establishment algorithm. We can obtain a simple equation for the delay bound test since the service times at the virtual channel operator is a constant.

Without loss of generality, assume that the set of channels considered in the delay bound test are $1 \,..\, K$ with $D^v_1 \leq D^v_2 \cdots D^v_K$, with $K \leq n$. Then, for $1 \leq j \leq K$ we need that:

$$D^e_j \geq t_q \, , \tag{C10}$$

with the appropriate t_q. In this situation, the number of packets before a packet on the j^{th} channel can only be j.

The sufficiency of these conditions follows from the correctness of the establishment scheme.

---

[3] The reader may have noted that the inclusion of t_o in the definition of t_s is just for the sake of completeness. Since no packet can be given to the physical channel operator at an interval smaller than t_o, it is unnecessary for any physical channel to have a value of $x_{\min}$ lower than t_o.

4. The Calculus Of Parameters

In the event that virtual channel establishment failed for all the physical channels available to the required destination, the virtual channel manager would like to translate the requirements of the new virtual channels into the requirements of a new physical channel. This physical channel must satisfy the tests mentioned in the previous section. We adopt the same notation for the parameters of virtual and physical channels as in the previous section.

For x^p_{ave} and I^p the least stringent values appear to be:

$$x^p_{ave} = 1/(\sum_{i=1}^{n} 1/x^v_{ave,i}) . \tag{F1}$$

$$I^p = \max_i I^v_i . \tag{F2}$$

Similarly,

$$t^p = \max_j t^v_i . \tag{F3}$$

In order to determine the deadline required of the physical channel, we must know the value of $x^p_{\min}$, since the calculation of physical channel's performance is dependent on it. Assuming it is known, we can use the following formula: with the same notation as in (C10),

$$D^p = \min_j D^v_j - j x^p_{\min} . \tag{F4}$$

The value of $x^p_{\min}$ depends on the type of channels being requested. If all the virtual channels are best-effort, we need not use (C8) or any special conditions and simply take it as the smallest of the virtual channels' minimum inter-arrival times. This is because we do not really care if the best-effort delay bounds are violated.

On the other hand, if all the virtual channels are deterministic, then we must take care to satisfy condition (C8). The maximum rate at which the virtual channel operator can send packets is one per t_o time units. It is therefore unreasonable to ask for a physical channel with a $x_{\min}$ lower than t_o. So we can assume that $x^p_{\min}$ is the value used for t_s in condition (C8). We use (C8) to calculate the resulting $x^p_{\min}$, which is $1/\sum_i 1/x^v_{min,i}$. If this value (i.e. $1/\sum_i 1/x^v_{min,i}$.) is less than t_o, then the set of virtual channels has to be broken into two or more smaller subsets and a different physical channel used for each subset. We can be assured that this will also satisfy (C1), since it has a peak arrival rate ($1/x_{\min}$) which exceeds that of any virtual channel.

When the virtual channels are all statistical, we may want to request a statistical or a deterministic physical channel. A statistical channel request is more likely to get accepted. We need to compute the value of both Z^p and $x^p_{\min}$. But we have only one equation for them, namely (C9). Obviously, more than one possible solutions exist.

Let us examine (C9) in some more detail. Figure 5 shows the typical shape of the probability distribution function used in that equation. [4] It shows the probability of virtual node saturation (i.e. the left hand side of (C9)) for different values of $x^p_{\min}$. The vertical axis can thus be taken as the possible values of Z^e. The only region of this graph that we can consider are the ones satisfying (C1) and (C6). In the figure, suppose all points to the left of B satisfy (C1) and all points above A satisfy (C6).

After selecting a point in the valid region of the graph, we can determine the values of $x^p_{\min}$ and Z^e. To obtain Z^p, we divide the maximum among Z^v_i by Z^e. It is not clear if a physical channel corresponding to point A or B has a better chance of getting accepted than one corresponding to point C. A channel corresponding to point C has a smaller value of the minimum inter-arrival time, but a more stringent probability bound than a channel corresponding to point A or B. In the

---

[4] The method of evaluating these probabilities can be found in [Fer89a].

statistical test for physical channels (see section 2), a higher value of $x^p_{\min}$ and a lower value of Z^p make the channel more likely to be accepted.

A simple scheme to obtain the combined minimum inter-arrival time for the physical channel is to use the smallest $x^v_{\min}$ to determine the right hand side of (C9). If this turns out to be smaller than the Z of a virtual channel, we find out the required minimum inter-arrival time for this bound, and request a deterministic channel.

As an example consider 3 virtual channels with these specifications

Virtual Channel Requirements			
number	$x_{\min}$	x_{ave}	D
I	10	40	25
II	20	40	30
III	25	40	35

If all the channels are best-effort, then we can use 10 as $x^p_{\min}$, and probably 25 as the delay required for the channel. If they are all deterministic, we should use a $x^p_{\min}$ of 1/(1/10+1/20+1/25), which is about 5 time units. The delay required by the channel is the minimum of (25-5, 30-10,35-15) which turns out to be 20 time units. We assume that t_o is smaller than 5 time units, and we do not need to break up the channels into smaller subsets.

If the channels were all statistical, with Z of 0.8,0.75 and 0.7, then we need to determine the value of Z^p and $x^p_{\min}$ together. The graph to be used for this solution is shown in Figure 5. We can use any pair between the points C and A. If we decided to use point C, we would request a statistical physical channel with Z^p of 0.8 / 1.0 and $x^p_{\min}$ of 5, while if we chose point A, we require a channel with $x^p_{\min}$ of 10 but Z^p of 0.8/0.8 i.e. 1 (or a deterministic physical channel). Which of the two is better is difficult to say at this moment.

A simple modification of the statistical solution will also work when multiplexing both deterministic and statistical channels over a deterministic physical channel.

5. Operators

The multiplexing scheme mentioned in the previous sections, if implemented in a simple-minded way, requires flow control and scheduling to be done twice for each packet to be transmitted, once by the virtual channel operator and then by the physical channel operator. Although the overhead of the scheme is small [Fer89b], it is doubled unnecessarily by this. Furthermore, the virtual channel operator needs to hold packets that arrive at an interval less than that permitted by the physical channel. This may lead to a situation where the network is idle but the virtual channel operator is holding some packets.

The above problems can be eliminated by close cooperation between the physical and virtual channel operators. Since multiplexing can only be done at the source, it requires the network layer (physical channel operator) to treat packets originating at the local site specially. If it is possible to merge the virtual channel operator with the physical channel operator, a scheme which requires flow control and scheduling to be done only once can be obtained. Further simplicity can be gained if we assume that all the virtual channels are of the same type.

As mentioned before, the only essential component of virtual channel operator is the mapping process. After this, we have the packet to be transmitted and two channel descriptors, one for the physical channel and the other for the virtual channel.

The