Title: BROADCAST OVERLAY PROTOCOL METHOD AND SYSTEM

A transmission method wherein a broadcast stream is superimposed over a set of one or more primary data streams by embedding delay-insensitive data packets of the broadcast stream during periods of inactivity in each primary data stream. The delay-insensitive data packets comprising the broadcast stream may be transmitted out-of-sequence relative to each other and out-of-band relative to the primary data stream. At least some of the delay-insensitive data packets are assigned acceptable error-tolerance levels, and a given delay-insensitive data packet has an optimum size and frequency of retransmission determined by a statistical decomposition of a data-time behavior of the primary data stream. The method has particular applicability in providing broadcast content (e.g., advertising streams) to World Wide Web (WWW) clients connected to an Internet service provider (ISP).
FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>Code</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Albania</td>
</tr>
<tr>
<td>AM</td>
<td>Armenia</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
</tr>
<tr>
<td>AZ</td>
<td>Azerbaijan</td>
</tr>
<tr>
<td>BA</td>
<td>Bosnia and Herzegovina</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d’Ivoire</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
</tr>
<tr>
<td>CU</td>
<td>Cuba</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
</tr>
<tr>
<td>FR</td>
<td>France</td>
</tr>
<tr>
<td>GA</td>
<td>Gabon</td>
</tr>
<tr>
<td>GB</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>GE</td>
<td>Georgia</td>
</tr>
<tr>
<td>GH</td>
<td>Ghana</td>
</tr>
<tr>
<td>GN</td>
<td>Guinea</td>
</tr>
<tr>
<td>GR</td>
<td>Greece</td>
</tr>
<tr>
<td>HU</td>
<td>Hungary</td>
</tr>
<tr>
<td>IE</td>
<td>Ireland</td>
</tr>
<tr>
<td>IL</td>
<td>Israel</td>
</tr>
<tr>
<td>IS</td>
<td>Iceland</td>
</tr>
<tr>
<td>IT</td>
<td>Italy</td>
</tr>
<tr>
<td>JP</td>
<td>Japan</td>
</tr>
<tr>
<td>KR</td>
<td>Kenya</td>
</tr>
<tr>
<td>KG</td>
<td>Kyrgyzstan</td>
</tr>
<tr>
<td>KP</td>
<td>Democratic People’s</td>
</tr>
<tr>
<td>KR</td>
<td>Republic of Korea</td>
</tr>
<tr>
<td>KZ</td>
<td>Kazakhstan</td>
</tr>
<tr>
<td>LC</td>
<td>Saint Lucia</td>
</tr>
<tr>
<td>LI</td>
<td>Liechtenstein</td>
</tr>
<tr>
<td>LK</td>
<td>Sri Lanka</td>
</tr>
<tr>
<td>LR</td>
<td>Liberia</td>
</tr>
<tr>
<td>LS</td>
<td>Lesotho</td>
</tr>
<tr>
<td>LT</td>
<td>Lithuania</td>
</tr>
<tr>
<td>LU</td>
<td>Luxembourg</td>
</tr>
<tr>
<td>LV</td>
<td>Latvia</td>
</tr>
<tr>
<td>MC</td>
<td>Monaco</td>
</tr>
<tr>
<td>MD</td>
<td>Republic of Moldova</td>
</tr>
<tr>
<td>MG</td>
<td>Madagascar</td>
</tr>
<tr>
<td>MK</td>
<td>The former Yugoslav</td>
</tr>
<tr>
<td>ML</td>
<td>Mali</td>
</tr>
<tr>
<td>MN</td>
<td>Mongolia</td>
</tr>
<tr>
<td>MR</td>
<td>Mauritania</td>
</tr>
<tr>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>MX</td>
<td>Mexico</td>
</tr>
<tr>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>NZ</td>
<td>‘New Zealand</td>
</tr>
<tr>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>SG</td>
<td>Singapore</td>
</tr>
<tr>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>TJ</td>
<td>Tajikistan</td>
</tr>
<tr>
<td>TM</td>
<td>Turkmenistan</td>
</tr>
<tr>
<td>TR</td>
<td>Turkey</td>
</tr>
<tr>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>UA</td>
<td>Ukraine</td>
</tr>
<tr>
<td>UK</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td>VN</td>
<td>Viet Nam</td>
</tr>
<tr>
<td>YU</td>
<td>Yugoslavia</td>
</tr>
<tr>
<td>ZW</td>
<td>Zimbabwe</td>
</tr>
</tbody>
</table>
BROADCAST OVERLAY PROTOCOL METHOD AND SYSTEM

This application is based on U.S. Provisional Application Serial No. 60/019,680 filed June 13, 1996.

BACKGROUND OF THE INVENTION

5 Technical Field

This invention relates to transmission of data over preexisting data channels and is especially applicable to apparatus and systems which transmit and receive broadcast data in unused spaces of these channels. Such systems can be of point-to-point in nature (such as that found in Internet infrastructure) or any other network topology.

Description of the Related Art

Networks deliver data through channels that are implemented with a wide variety of media and technologies. Networks can take on many forms but will always be comprised of at least three essential components: a transmitter, a medium, and a receiver. Typically, these networks are designed to use the available data-carrying capacity (also known as "bandwidth") in an efficient manner. However, networks seldom use all of the available bandwidth all of the time. The reasons for this are many. Installed networks are often designed with excess carrying capability. Other networks may not have excess carrying capacity but remain inefficient due to usage patterns or transmission technologies.

The most prevalent mode of transmitting data is currently point-to-point in which data leaving a transmitter is intended for only one receiver. As networks support more individuals and virtual connections (point-to-point paths), the benefit of broadcast (one-to-many) or multicast (many-to-many) is often being overlooked even though the network infrastructure could easily support such modes of data transmission. This is not altogether difficult to understand as network technologies

1
have been evolving to support the ever-increasing demands placed by client-server applications and Internet (e.g., World Wide Web (WWW)) applications which are conceptually point-to-point.

Because of the emphasis on point-to-point transmission and usage patterns, excess capacity that can be used to deliver useful non point-to-point data goes largely under-utilized. For example, if one takes a look at usage patterns of a typical user of a dialup (POTS) SLIP/PPP connection to an ISP (Internet Service Provider), one will see that there are extended periods of time in which no data transfer is taking place. In fact, often there is an inactivity "time-out" for which the ISP will disconnect the user if a period of inactivity persists beyond a certain threshold. This known approach is usually dictated by the fact that the ISP often charges a flat-rate or a nominal fee-per-minute, which makes it unviable for ISPs to allow indefinitely inactive connections. Moreover, often there is a hardware limitation that restricts the number of connections which an ISP can support at any one time. Further, there may also be corresponding processing limitations.

Much of the information sent through such a system is demand-driven, i.e., it is pulled through the network. It is largely the lack of pushed data that prevents one from using periods of inactivity. Ideally, one would send pulled data when it is available and transmit pushed data when no pulled data is ready. The problem with such an approach is that, currently, there is little demand for transmitting pushed data. This is primarily due to the fact that most users are only interested in data which they have requested directly, while pushed-data is typically not generated for a specific user or in response to a real-time query. In fact, the most efficient use of pushed-data would be to send broadcast data to many users.
One problem is that, in many current network systems, broadcast data can be sent no more efficiently than individual point-to-point data. "Broadcast" data on these networks are simply sent much like any other data except it is sent repeatedly to multiple users. True broadcast typically can only occur on "bus-topology" networks, i.e., networks in which a signal placed on the network is available to all receivers. This is the case for networks that utilize such technologies as Ethernet and wireless, for instance.

This is not the case for ISPs currently, however. In order for most ISPs to send a "broadcast" packet to its users, the ISP has to send the same packet repeatedly to all of the connected users by basically opening a virtual connection to each user. This is not very resource efficient, but is the only practical way with the current standard hardware and software.

The majority of today's ISPs supply service through POTS or ISDN, both of which are virtual-circuit based technologies, i.e., a dedicated "virtual circuit" is established between the two ends of the channel. Such a circuit can actually correspond very closely to a real physical circuit since both POTS and ISDN have a separate twisted-pair between each customer and the central office switch. In a network based on POTS or ISDN, true broadcast is not built into the system. On the other hand, ISPs that provide service through cable-modems, for instance, will be able to provide broadcast data for very little additional cost.

The problems of bandwidth under-utilization and low-cost broadcast ability on traditional POTS/ISDN infrastructure are of particular relevance to the present invention. These problems are interrelated in as much as low-cost broadcast ability could be used to better utilize total available bandwidth. The present invention seeks to solve both
problems, as well as to provide a complete system in which newly-available broadcast capacity becomes an attractive supplement to the pull-only data currently available. This last feature is important because, though the current trend in networks will increase the availability of bus-topology networks, the attendant broadcast ability will go under-utilized unless a system is created to make broadcast data more usable and attractive.

The present invention addresses these problems associated with the prior art.

SUMMARY OF THE INVENTION

According to the present invention, broadcast data is made feasible over traditional circuit-based networks as well as over non-bus topology networks through the introduction of a novel data format and protocol. This protocol is referred to herein as a "broadcast overlay" because it enables a secondary broadcast stream to be "overlaid" or embedded within a primary data stream. The format and protocol thus enables the insertion of broadcast data in periods of inactivity with little or no negative impact on the delivery of primary point-to-point data.

Thus, it is a principal object of this invention to provide a broadcast protocol that takes advantage of current bandwidth under-utilization.

It is another primary object of this invention to define a highly error-tolerant and useful transmission protocol for "complementing" an existing underlying transmission protocol, such as (SL)IP, PPP or HDLC, without having to modify the existing protocol.

It is still another object of this invention to provide an overlay or "adjunct" protocol for use with an existing or "underlying" protocol, wherein the adjunct protocol is designed so that its own packets are interruptible by the underlying protocol within the minimum amount of
latency to ensure that the underlying protocol transmission performance is not impacted.

Yet another object of this invention is to overlay data packets conforming to a overlay protocol on top of pre-existing protocol streams, but preferably only during at least some periods of underlying protocol inactivity.

It is yet another object of this invention to allow a single overlay protocol packet stream to be generated and superimposed on an arbitrarily large set of pre-existing data streams. This invention thus facilitates broadcast-mode transmission, even though the underlying physical medium is not, by convention, a broadcast medium.

It is a further object of the present invention to allow for a single broadcast data stream to be generated and, with minimum additional resource and cost-per-user, send this information to a large number of users. The invention thus realizes the benefits of a broadcast medium, namely maximizing data distribution at minimum per-user costs.

More generally, it is an object of the present invention to exploit the fact that typical use patterns of an Internet user creates periods of inactivity during which additional information (broadcast or otherwise) may be sent to the user. The overlay protocol (OP) is designed to exploit these inactivity periods.

It is a further object of this invention to implement the broadcast overlay protocol in conjunction with Internet Service Provider (ISP) delivery system for World Wide Web users.

The invention provides a transmission method wherein a broadcast stream is superimposed over a set of one or more primary data streams by embedding delay-insensitive data packets of the broadcast stream during periods of inactivity in each primary data stream. The delay-insensitive data packets comprising the broadcast
stream may be transmitted out-of-sequence relative to each other and out-of-band relative to the primary data stream. At least some of the delay-insensitive data packets are assigned acceptable error-tolerance levels, and a given delay-insensitive data packet has an optimum size and frequency of retransmission determined by a statistical decomposition of a data-time behavior of the primary data stream. The method has particular applicability in providing broadcast content (e.g., advertising streams) to World Wide Web clients connected to an Internet service provider.

These and other objects of the invention are provided in a method of transmitting data in a communications network having a user connectable to a data center via a channel. In a representative embodiment, the user is a client machine, the data center is an Internet service provider (ISP) and the communications network is the Internet. The method begins by transmitting a primary data stream from the data center to the user, the primary data stream comprising a plurality of data packets conforming to a first given protocol and characterized as having periods of inactivity. The given protocol may be, among others, PPP, SLIP and/or HDLC. As the primary data stream is being transmitted from the data center to the user, the method embeds into the primary stream a secondary "broadcast" stream comprising a plurality of delay-insensitive data packets conforming to a second given protocol -- the overlay protocol of this invention. The secondary stream is embedded into the primary data stream without substantially affecting latency of data packets comprising the primary data stream. The secondary stream is then reconstituted at the user's client machine.

According to the invention, the secondary "broadcast" stream is embedded into the primary stream by embedding the delay-insensitive broadcast data packets during the periods of inactivity in the primary
data stream. The delay-insensitive data packets comprising the secondary stream may be transmitted in the primary stream out of sequence or particular data packets may be transmitted multiple times within the primary stream.

Thus, according to the invention, the secondary broadcast stream embedded in the inactivity periods of the primary data stream is "interrupted" as necessary so that the primary data stream itself is not delayed. This may be achieved by defining the overlay protocol so as to exclude a given flag sequence of the primary data stream. In this way, the embedding of the secondary stream is interrupted upon occurrence of the given flag sequence. Where the primary data stream is transmitted via a protocol that does not include flag sequences, interruption of the broadcast stream (without impacting the latency of the primary data stream) may be achieved by defining a start sequence in the overlay protocol that is restricted from occurring in a header of the primary stream.

In a particular application, the primary data stream includes information retrieved from a web site connected to the Internet service provider and the secondary broadcast stream includes a stream of content (e.g., product/service advertisements) that is then displayed on a display interface of the client machine.

The invention also describes a transmission system located at a service provider for transmitting data over a set of one or more connections to a set of one or more clients, respectively, the service provider connected to a plurality of servers. The transmission system comprises first means responsive to a request from a client for communicating with one of the servers and receiving information to be transmitted to the client. A second means responsive to the first means is provided for transmitting the information to the client in a primary data
stream, the primary data stream comprising a plurality of data packets conforming to a given protocol and characterized as having periods of inactivity. The system further includes a third means responsive to the second means for embedding into the primary data stream a secondary broadcast stream as the primary data stream is transmitted to the client. The secondary stream comprises a plurality of delay-insensitive or "interruptible" data packets conforming to an "overlay" protocol. The secondary broadcast stream is embedded into the primary stream without substantially affecting latency of the delay-sensitive data packets comprising the primary stream. In a representative embodiment, the transmission system is an ISP delivery system, in which case the request is an HTTP request, the server is a web site and the information is a web page. In this example, the secondary broadcast stream includes a plurality of advertisements.

The foregoing has outlined some of the more pertinent objects and features of the present invention. These objects should be construed to be merely illustrative of some of the more prominent features and applications of the invention. Many other beneficial results can be attained by applying the disclosed invention in a different manner or modifying the invention as will be described. Accordingly, other objects and a fuller understanding of the invention may be had by referring to the following Detailed Description of the Preferred Embodiment.

**BRIEF DESCRIPTION OF THE DRAWINGS**

For a more complete understanding of the present invention and the advantages thereof, reference should be made to the following Detailed Description taken in connection with the accompanying drawings in which:
Figure 1 is a block schematic diagram of a section of a transmission network in which the present invention is implemented;

Figure 2 is a diagram illustrating a representative packet format of the overlay protocol (OP) of the present invention;

Figure 3 is a diagram illustrating a known HDLC packet format;

Figure 4 is a diagram illustrating a sample HDLC packet stream;

Figure 5 is a diagram illustrating a sample OP packet stream according to the present invention;

Figure 6 is a diagram illustrating a sample broadcast overlay protocol stream which is created by overlaying the OP packet stream of Figure 5 onto the HDLC packet stream of Figure 4;

Figure 7 illustrates a simplified technique for generating the broadcast overlay protocol stream of Figure 6;

Figure 8 illustrates a simplified technique for embedding a secondary broadcast stream into multiple user data streams;

Figure 9 illustrates an implementation of the broadcast overlay protocol in an Internet service provider (ISP) delivery system to utilize the downtime in a typical Internet browsing session to download broadcast content;

Figure 10 depicts a Web browser of a client machine displaying broadcast data supplied to the client machine using the inventive protocol;

Figure 11 is a simplified block diagram of band extractor which is used according to the present invention to determine the frequency of re-transmission of a given broadcast data packet through a statistical decomposition of a data-time behavior of the primary data stream; and

Figure 12 is a block diagram representing various functional "layers" of a client-side system supported at a user's machine.
DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 illustrates a representative system architecture in which the present invention is implemented. The system comprises three (3) major classes of entities: a "user" 10, a "data-center" 12 and a "broadcast-client" (or server) 14. A user 10 may be a client machine (e.g., a personal computer) connected to a computer network in a known manner. The network may be any known computer network such as the Internet, an intranet, a wide area network (WAN) or local area network (LAN). In the illustrative embodiment, the network is the Internet. A "data-center" 12 is the set of machines and related equipment whose responsibility is to supply a set of "users" with access to the network, such as the Internet, as depicted in Figure 1, and broadcast data. A representative data center 12 is an Internet Service Provider (ISP). The data-center 12 typically provides services to "broadcast clients" 14 which allows it to submit and control broadcast data as well as to submit "remote jobs" and collect information from users. A machine that is used as a broadcast client 14 may also be a user and vice-versa. Figure 1 shows these components as separate entities for the sake of simplicity. A broadcast client may support a server 17 supporting HTML documents, sometimes referred to herein as a web site.

The data-center 12 comprises several major components: the broadcast server 16, a response server 18, a mux/demux 20, and the Internet gateway 22. The Internet gateway 22 allows users to send and receive IP packets to and from the Internet 24, and this component performs ancillary functions such as routing and IP address assignment. This is the primary service provided by ISP (Internet Service Provider) data-centers. The client machine is any personal computer or workstation, such an Intel x86-based processor running an operating system (such as Microsoft Windows '95/NT), and supporting a
conventional Web browser (such as Netscape Navigator 3.0 or higher or Microsoft Explorer 3.0 or higher). The Web browser provides a graphical user interface for supporting the retrieval and display of Web documents via the Internet's World Wide Web multimedia information retrieval system.

The data-center 12 supplies SLIP/PPP-like service to the user in a conventional manner. Moreover, according to the present invention, the data center also allows a broadcast data stream to be sent and user-responses to be received, both out-of-band relative to the data in the IP or any of the PPP protocols. Such data can be sent out-of-band and still remain compatible with protocols like PPP, HDLC and SLIP, by carefully avoiding the use of the so-called "flag codes" associated with the start and end of packets in these protocols, as will be seen.

This out-of-band information is encoded with a novel protocol introduced in this invention. This protocol, called Overlay Protocol (or OP), differs from other protocols as it is designed to be highly error-tolerant and useful even when OP packet integrity cannot be guaranteed. This protocol is not designed to be a standalone protocol, but rather is to be used with other preexisting protocols, such as (SL)IP, PPP, or HDLC, without having to modify these protocols.

The OP protocol is designed so that its own packets can be interrupted by the underlying protocol with the minimum amount of latency to insure that underlying protocol transmission performance is not impacted. OP packets are overlaid on top of preexisting protocol streams and are only transmitted during periods of underlying protocol inactivity.

Figure 2 shows the packet format defined by the OP protocol. It contains a FLAG field 30, an OBJECT# field 32, a PACKET# field 34, a TYPE field 36, the DATA field 38, an FCS field 40, an ERRTOL field 42,
an LEN field 44 and a FLAG field 46. The FLAG fields 30 and 46 define the start and end of the given packet. The OBJECT# and PACKET# fields provide object and packet definitions. The TYPE field is a special field that facilitates defining the error tolerance level of the data packet. By way of example, the possible types include the following: 1 = binary, 2 = text, 3 = image, 4 = sound, 5 = video, and so on. As will be described below, the OP data packets may have a defined error tolerance, which is specified by the ERRTOL field. In accordance with this invention, the error tolerance of the packet can be set so that even in cases of errors in the packet, the packet will still be used. The error tolerance level is the variable length field ERRTOL whose length (defined by the LEN field) depends on the OP TYPE field 36. The error tolerance levels are preferably encoded so that only certain values are valid, for example, the value 10,000 is associated with an error tolerance of 1%, the value 20,000 is associated with an error tolerance of 2%, and so on. This is done so that the error tolerance level can be extracted with a high degree of confidence for its data integrity. For data types that cannot tolerate any errors (e.g., binary or text), ERRTOL is typically set to zero. For data types that can tolerate some error (e.g., sound or video), ERRTOL can be set to the user-defined tolerance levels.

Figure 3 shows the format of a conventional HDLC packet. This is included as an example of an underlying protocol on which the OP packets are overlaid according to this invention. Preferably, the OP packets do not replace or preempt any of the underlying protocol's data. Instead, such packets are only transmitted during periods of inactivity in a way that minimally impacts the preexisting data stream. The following example illustrates this concept:

Figure 4 represents a typical HDLC packet stream. Notice that this stream contains periods of inactivity 50 during which no information
is being transmitted. **Figure 5** represents a typical OP packet stream according to the present invention. It is preferred (although not required) that the OP stream is continuously broadcasting information, i.e., there are no periods of inactivity. It should also be noted that the packets are not necessarily in order and that particular packets may be transmitted multiple times. Thus, for example, packet **OP3** is transmitted both before and after the first transmission of packet **OP4**, which itself is transmitted twice. The packet **OP3** is also transmitted a third time, between the second transmission of packet **OP4** and the first transmission of packet **OP5**. Of course, this illustration is merely representative.

To achieve the objects of the present invention, the sample OP packet stream of **Figure 5** is overlaid on the underlying protocol stream of **Figure 4**. The resulting broadcast overlay protocol stream (which is a single stream) is illustrated in **Figure 6**. The stream in **Figure 6** is thus created by superimposing (or "overlaying") the OP packet stream on top of the HDLC packet stream during (at least some) periods of HDLC inactivity. It should be noted that, typically, not all of the OP packets are transmitted in their entirety. This is because the inventive transmission method does not delay or hold-off HDLC packet transmission to allow an OP packet to be transmitted in its entirety. In other words, the OP packet stream is interruptible by the underlying packet stream.

**Figure 7** illustrates a convenient mechanism for embedding the OP packet stream (sometimes referred to as a secondary or "broadcast" stream) into the underlying packet stream (sometimes referred to as the primary or "user" data stream since this is the stream that includes the data requested by the user). As seen in this drawing, a Virtual OR operator **52** takes the primary data stream **54** as one input and the single broadcast data stream **56** as a second input and generates an
output data stream 58 with the embedded broadcast information. The broadcast information may comprise any type of content including, without limitation, graphics, text, animation, video, sounds, programs and combinations of the above, as well as appropriate formatting information for controlling a Web browser (or other GUI) to display or otherwise output the broadcast information. One convenient implementation is a display of scrollable animated or static advertisements and/or videos in a dedicated content window or enlarged scrollbar area of a Web browser display interface.

The "Virtual OR" operator is an extremely simple (low-computational expense) operation as defined by the following:

\[
\begin{align*}
\text{BDS}(t) \text{ VOR } \text{UDS}(t) &= \text{BDS}(t) \text{ if } \text{BDS}(t) \text{ is non-null} \\
\text{and} \\
\text{BDS}(t) \text{ VOR } \text{UDS}(t) &= \text{UDS}(t) \text{ if } \text{BDS}(t) \text{ is null}
\end{align*}
\]

where:

VOR is the Virtual-OR operator,
BDS(t) is the broadcast data stream at time "t" and
UDS(t) is the user data stream at time "t".

It should be appreciated that this operator is "stateless" and does not require storage (memory) to be implemented. Because the VOR operator can be implemented economically, this technique scales very easily and inexpensively. Thus, an ISP for example may increase the audience of the broadcast stream for little or no additional incremental cost.

Figure 8 illustrates the simple scalability of this system wherein multiple user streams 60 (at least some of which are different) receive the same broadcast stream 62 through the use of a plurality of VOR operators 64. This technique is advantageous in an ISP delivery system implemented at the data center. In particular, conceptually, each of the
user data streams corresponds to an individual user of a Web browser at a client machine. Each such user is making requests to Web servers over the WWW using an underlying protocol transmission. The "inactivity" periods of these underlying protocol transmissions are then exploited according to the present invention by embedding the same broadcast stream content into those underlying transmissions.

While the above diagram (and others presented herein) shows only the user-data stream being combined with a broadcast data stream, this invention does not limit embodiments to this purpose only. This invention exploits the fact that typical usage patterns leaves periods of inactivity during which additional information (broadcast or otherwise) can be sent to the user.

The communications protocol of the present invention provides significant benefits. One particular use is an Internet application to exploit the "downtime" in a typical Internet browsing session. The present invention enables an ISP to broadcast "push" content to its customers (i.e., clients) without interrupting the normal flow of Internet communication. In a preferred embodiment illustrated in Figure 9, content for distribution is loaded onto and administered from a broadcast server (see Figure 1) at the ISP site. Software running at the data-center identifies the downtime in communication between the ISP and the user. The broadcast data is then sent to the user using the inventive overlay protocol. As described above, the protocol works in conjunction with the IP protocol without interrupting the IP transmissions. The ISP broadcast data is then rendered at the user's computer, e.g., in a unique content window as illustrated in Figure 10. The software at the data-center may collect individual usage information and then analyze this information to optimally target advertisements for the user.
A large amount of the simplicity found in this data overlaying system is due to the fact that the overlaid OP packets may be interrupted at any time. To actually implement this feature, it is necessary to detect when an OP packet has been interrupted. One way to do this would be to mark the interruption of an OP packet with a sequence that identifies the end of OP and the start of underlying protocol transmission. Assuming this sequence was one byte long (the shortest practical sequence), one would have to delay and buffer the transmission of the underlying protocol for the length of this one-byte sequence.

This buffering introduces some additional latency and resource requirements that are not always desirable. To obviate such buffering, other, more preferred techniques may be implemented to interrupt the OP packet stream. One preferred approach is to define the OP protocol in such a way as to eliminate all occurrences of the flag-sequence of the underlying protocol in the OP data stream. In this way, the interruption of OP protocol data transmission is recognized upon the reception of the underlying protocol's flag-sequence.

For example, consider a system in which OP is superimposed on HDLC. Referring back to Figure 3, it should be recalled that all HDLC packets are bracketed by its defined flag sequence. In the case of HDLC, its flag sequence is one byte long and consists of the binary sequence \texttt{0111 1110} (hexadecimal \texttt{0x7E}). According to the present invention, this sequence is restricted from occurring during OP transmission. If the value \texttt{0x7E} needs to be transmitted anywhere within the OP data stream, it is replaced by a sequence consisting of an "escape" code plus the value XORed with an appropriate non-zero mask. Any other value (especially the escape code) can then be
replaced with the escape code followed by the value XORed with the mask.

Alternatively, one may detect the resumption of OP protocol transmission upon reception of the trailing flag sequence. Certain protocols do not dictate the packet be bracketed by flag sequences but instead merely prefixed with the sequence. In fact, HDLC, allows the trailing flag-sequence to be omitted when it is followed immediately by another HDLC packet. This presents a potential problem in that, with such protocols, it is not easy (and sometimes quite difficult) to distinguish between OP data and a packet that has not been preceded by a closing flag-sequence.

The present invention solves this latter problem. In particular, upon the resumption of OP transmission in systems that overlay on top of protocols that do not require bracketed packets, the protocol requires that an OP start sequence be embedded immediately after the end of underlying protocol's frame. This OP start sequence must then be such that it cannot occur legally in the header of the underlying protocol's frame. The requirement to transmit this start sequence does not require the system to buffer the OP packet, nor does it introduce any latency in the transmission of underlying protocol. The reason for this is that there is still no requirement to delay the transmission of the OP packet since the protocol permits sending of partial frames. In other words, since there is freedom to start and stop the OP transmission anywhere it is most convenient, transmission may start without penalty after the start-sequence has been sent.

In the case of OP-over-HDLC, the OP start-sequence is a two-byte sequence of the form: aa:bb, where bb=0x7D (the HDLC escape sequence). The 'aa' sequence in this case is variable and is actually the value of the OP data stream that bb replaces. Thus, the second byte of
the OP start-sequence holds the actual flag value and the first byte holds the value that the second byte replaces. This technique is convenient because HDLC allows any value to occur in the address field of its frame. While the control field (the next field) only allows the value 0x03, the reception of the escape sequence 0x7D is not legal within the context of the HDLC's control field. Alternatively, one could define a similar three-byte sequence to replace the HDLC protocol field with a value that did not have its LSB set (which is another illegal condition as defined by HDLC).

At anytime upon the reception of the underlying protocol's flag-sequence, the OP protocol dictates that what follows be interpreted (preferably) as underlying protocol data. The reception of the underlying protocol's flag-sequence can occur at anytime including during the OP start-sequence. It should be appreciated that the novel format of the OP protocol thus allows it to adapt to nearly any underlying protocol.

However, the flexibility of the protocol is not just limited to its adaptability to other protocols. As illustrated above, OP does not impose a minimum or maximum receive unit (MRU), nor does it impose a sequence in which packets must be ordered. This novel feature allows the present invention to transmit the broadcast data in whichever way it is most appropriate. This adds tremendously to the value and utility of the invention by allowing the transmission behavior to be tailored specifically to the transmitted data and channel(s). This adaptability of transmission to the data and channel conditions greatly increases the effective throughput of a broadcast stream by tailoring transmission parameters to best match conditions and thus increasing the likelihood of successful reception of packets.

For example, take the case of a POTS channel through which a user is connected to the data-center. Suppose, this channel must go
through at least two switches and that there is a condition known as
clock-slip between these two switches, which results in a burst of noise
approximately every two seconds. If the OP data packet size were set
so that it took approximately two seconds to transmit each packet, it
could potentially take a very long time during which the packet was
repeated before the packet could be successfully reconstructed.

Although OP allows for partial transmission of packets and the re-
assembly of these partial packets into complete packets, this process
can take many repetitions in cases like the one above. This is still better
than many protocols for which situations such as this one are not
resolvable; however, this solution presented above may not always be
optimal. The present invention takes two approaches to the problem of
interference (due either to noise or underlying protocol activity). This
feature of the present invention is now described.

The first is to define the error-tolerance level of OP data packets;
the second is to perform what is referred to as adaptive-directed,
statistical-decomposition (ADSD) of the channel.

The error-tolerance level of a packet can be set so that even in
cases of errors in the packet, the given packet may still be used. The
error tolerance level is the variable length field whose length depends on
the OP TYPE field. The error-tolerance level is specifically encoded into
the OP ERRTOL field so that only certain out of a great many values are
valid. This is done so that the error tolerance level can be extracted with
a high-level of confidence. If this were not done, it would be possible
that an error itself was responsible for changing the error tolerance level
of a packet. To further safeguard the integrity of the system, only certain
types of packets (specified by the OP TYPE field) can legally have a
non-zero error tolerance. For example a packet containing an
executable is not permitted to have an encoded non-zero ERRTOL value.

The error tolerance of the packet can be defined to be non-zero for a variety of reasons. The data contained in the packet may be encoded in a way that allows errors to be detected and corrected. Alternatively, the data can be deemed to be error tolerant. This is especially useful in the case of progressive transmission of media in which errors in the higher-frequency components can be tolerated. It is even possible to low-pass filter packets in which errors have been detected in order to minimize the effects of noise in the data. This approach could be applied to both low-frequency and high-frequency components of transmitted media.

Adaptive-directed, statistical-decomposition (ADSD) is another approach that is implemented in this invention (in addition to defining individual packet error tolerance) to deal with interference. This function is performed by a band extractor (preferably implemented in software) that is illustrated in Figure 11. The band extractor includes a number of functional components, a DT-space data collector 70, an autocorrelation function 72, an exponential weighted averager 74 and a DF band selector 76. In general, the band extractor facilitates a statistical decomposition of a "data-time" behavior of the primary data stream which is then used to determine the frequency of re-transmission of given data packets constituting the broadcast data stream. This operation will now be described.

ADSD analyzes interference (whatever its source) as a complex periodic signal that can be decomposed into a collection of statistically-relevant components. For instance, in the example above, the line noise created by the clock-slip between the two switches would be analyzed by the ADSD module of this invention and decomposed into statistically-
significant noise components (in this example, there is only one statistically significant noise component at 0.5Hz).

Time-domain behavior of the interference is not as important as data-time-domain behavior or DT-space (or Data-Time space). To project a signal into DT-space, the DT-space data collector 70 is used to sample the real signal in the time-units of the protocol of interest. This is a relatively straightforward process for low-level protocols such as RS-232, where the projection into DT-space is mostly a matter of scaling the time-axis by the transmission rate. For higher-level protocols which contain complex frames, the projection into DT-space can be approximated by the collector 70 by scaling the time-axis by the frame-size.

In order to determine what is happening in the frequency domain corresponding to the DT-space signal, an autocorrelation may be performed of the DT-space signal by the autocorrelation function 72 to estimate what is happening in the DF-space. In order to place more emphasis on more-recent events and trends, the result of the autocorrelation can be taken and passed through the exponential weighting averaging function 74. Peaks represent recent significant periodicity in the interruption of the data signal, wherein dips represent few interruptions at the given periodicity. Preferably, the broadcast data is sent during this time since it represents the periodicity at which there is the least chance of being interrupted.

According to the present invention, the optimum size and frequency of re-transmission of broadcast data packets is preferably determined by the statistical decomposition of the data-time behavior of the primary data stream into which the broadcast data packets are to be embedded.
In particular, the information culled from the band extractor shown in Figure 11 is collected by the broadcast data stream transmitter, which then uses this data to determine how to break up the broadcast packets (which thus determines their optimum size) and how often to re-transmit them. An implementation of this function is now described. It begins by collecting a "successful" packet transmission rate as a function \( S(f,l,r,T) \) of data sampling frequency \( f \), package length \( l \), number of retries \( r \) and total sampling period \( T \). This step is performed by the collector 70. A spectrum estimation is then done from the \( S(f,l,r,T) \) data set using the autocorrelation function 72 (or some other suitable algorithm that may suit the operating environment). The routine then extracts the optimal successful packet transmission rate for the given transmission media. The optimal successful transmission is typically a quadruple of the \( f,l,r \) and \( T \) parameters. The optimal successful transmission parameters are then used to schedule a next transmission to the media. This process is repeated in a continuous feedback loop.

The band extractor works optimally on a single data channel, so for every data channel to the user a separate band extractor task must be running. This could result in a large amount of computation done at the data-center if this information were to be gathered at the broadcast site. Thus, preferably the band extractor is located at the user site (as embedded software) to significantly reduce the amount of computational power needed at the data-center.

The computational power needed for the band extractor is minimized by using the autocorrelation function instead of a more expensive Fourier-transform (which might be used as well if desired). The band extractor further minimizes the amount of computation by collecting the result of the autocorrelation function block and placing it into bins of decreasing size along the time (lag) axis. Furthermore, the
exponential averaging is done with the aid of look-up tables and the band selector is based on a simple heuristics and histogram mechanism supported in the DF band selector 76.

In addition to this band extraction mechanism, the invention envisions that the user (i.e., the client machine) indicates which packets it has successfully received so that the data-center may, through a simple histogram mechanism, determine the relative priorities of the packets. The combination of these two mechanisms ensures that there is enough information present to make an intelligent decision on which packets should be given higher priority and how often they should be transmitted. The data-center is also able to direct each of the band extractors running at the user sites to divide up the entire data frequency space into specific bands. This is used to minimize the amount of data that is received by the data-center since the band extractors only extract information that the data-center has requested. This technique is also used to minimize the amount of computation that the user site must expend because the program is only computing what is necessary (to complete the data-center specification). This ability to change how the data is decomposed into varying units is what is meant by "directed-adaptive" in the acronym ADSD.

Referring now to Figure 12, a block diagram is provided to illustrate the client-side system and the data flow between various functional components. It should be appreciated that not all of the components illustrated herein are necessary to facilitate use of the broadcast overlay protocol transmission method. Moreover, these components are preferably implemented in software using conventional programming techniques. As noted above, the client machine includes a processor, an operating system, a Web browser for displaying content comprising the broadcast stream.
The system comprises multiple layers, each of which provides novel and useful features which facilitate the providing of an integrated data broadcast environment. The lowest layer 80 defines the hardware interaction between communication devices in a way that facilitates multiplexed (i.e., broadcast) communications. This layer can be implemented with a combination of hardware, firmware and software solutions ranging from simple analog electronics devices, reprogramming telecommunications switches, high level protocols or any combination thereof. The next layer 82 supports the underlying communication devices different modes of operation (e.g., quiescent, interactive, broadcast, etc.), and enables the upper layers to recognize and set these modes. The next layer 84, on reception, separates the new data stream from the standard (i.e., the currently transmitted) information as well as overlays new data, on transmission, in a way to minimize (eliminate) latency to the preexisting data streams and impact on transmission performance of this stream. This operation has been described in detail above. Layer 84 comprising a filter 85 which processes the overlay protocol and data format. The filter monitors the incoming data stream and separates raw broadcast information from the other packet information. This raw data is processed and buffered in such a way that recovery of useful broadcast information can be made from partial broadcast packets by higher-level software.

To this end, the next layer 86 processes raw packets to make them easily usable by higher layers, as well as to expose hooks (application programming interfaces or "APIs") that enable implementation of protocol features such as security and progressive transmission. The next layer 88 authenticates and provides access to data services (which may be optionally) supplied by this system. This layer also exposes hooks that enable the creation of "virtual" IP
communications. The next layer 90 (also optional) provides higher level functionality to facilitate applet implementation by providing APIs for scheduling tasks for events and idle time. The next layer 92 supplies APIs and libraries for higher level applications to insure they are "safe" as well as well-behaved (e.g., easily uninstallable). The highest level 94 supplies APIs to control and enumerate broadcast streams and high-level functions associated with systems management (e.g., power-management, alarms and task scheduling).

In summary, the present invention provides a novel protocol and transmission method. According to the invention, a secondary "broadcast" stream is overlaid on a set of one or more underlying packet stream(s). The technique does not delay or hold-off underlying packet transmission to allow an OP packet to be transmitted in its entirety. There are three important benefits to this technique: (1) minimizing (or eliminating) latency of HDLC (or any other underlying protocol) packet transmission - i.e., HDLC packets go out at the same rate and time regardless of the presence of OP packets; (2) minimizing (or eliminating) any buffering associated with holding off the underlying protocol packets - i.e., there is no need to buffer HDLC packet data because the technique does not delay its transmission; and (3) simplifying system implementation and per-user computational and resource costs.

The third benefit may not be immediately apparent in the two-stream example given above, but it becomes apparent when examining the multi-stream case. Given several arbitrary streams of HDLC packets, the holding-off and delaying of the HDLC packets, e.g., in order to embed a secondary stream according to this invention, either imposes severe performance penalties on HDLC packet transmission or requires a high-level of per-user resources. The reason for this is that, in order to insure that every OP packet is received by each of the users (the
primary reason for delaying HDLC packets), upper level protocols would be required to stop the production of HDLC packets; alternatively, an indefinite number of HDLC packets and OP packets would have to be buffered.

The present invention simply allows a single OP packet stream to be generated and superimposed on an arbitrarily large set of preexisting data streams. In contrast, if one were to hold off packets to embed broadcast streams, e.g., creating a PPP-compatible OP protocol and the corresponding PPP OCP (OP Control Protocol), one would be in effect "individualizing" the broadcast stream for each user because, although the same data is being sent to each user, it is not being sent at the same time. This would not provide the full benefits of a broadcast data stream, namely, to maximize distribution of data and minimize per-user cost.

The technique of this invention allows for full broadcast-mode transmission, even though the underlying physical medium may not be traditionally thought of as a broadcast medium. This invention allows for a single broadcast data stream to be generated and, with minimal additional resource and cost-per-user, this stream is then sent to a large number of users. Thus, the technique realizes all of the benefits of a broadcast medium — to maximize data distribution and minimize per-user costs. The low per-user costs results from the absence of a complex buffering system, the absence of large amounts of memory for buffering user data, and a truly single broadcast stream system.

It should be appreciated that the invention also enhances the usability and value of aforementioned broadcast data by enabling application security, progressive-transmission, robust media, error tolerance, intelligent caching and site-emulation. Such features can be used independently of the data protocol/format and related features mentioned above. These features stand by themselves as broadcast
data may be made available independent of the first part of this invention.

The preferred embodiment of this invention includes the first feature when used with a non-bus network, and also contains the other features that enhance the usability of broadcast data regardless of its origin. The present invention includes a security module, but in addition to this module, this invention inherently offers a higher level security than is currently afforded over a similar network which provides only point-to-point services. The reason for this is due to the topology of embodiments of this invention which reduces the number of possible security "points-of-failure" between the user and secure objects.

The invention reduces these security loci preferably by moving the secure objects from a client site to the site of the data-center itself. Making these security objects "pushed-data" (i.e., broadcast) combined with the technology introduced by this invention makes it preferable to move said objects from the client (where they have been traditionally stored) to the data-center. This is dictated by security issues as well as the reduction of cost to the client associated with offloading this work to the data-center, as well as the low cost to the data-center using the present invention.

In addition to offloading data transmission responsibilities from the client site, this invention also enables the client to offload computation to the user's machine or data-center machine. The user's machine is the preferred location to offload computation, however, as one of the primary benefits of the present invention is to minimize the amount of computation needed to support broadcast mode transmission.

One of the preferred implementations of the invention is a computer program product in a computer-readable media, for example, a software application (a set of instructions or program code) in one or
more code modules which may, for example, be resident in the random
access memory of a computer. Until required by the computer, the set
of instructions may be stored in another computer memory, for example,
in a hard disk drive, or in a removable memory such as an optical disk
(for eventual use in a CD ROM) or floppy disk (for eventual use in a
floppy disk drive), or downloaded via the Internet or other computer
network. Thus, the present invention may be implemented as a
computer program product for use in a computer. In addition, although
the various methods described are conveniently implemented in a
general purpose computer selectively activated or reconfigured by
software, one of ordinary skill in the art would also recognize that such
methods may be carried out in hardware, in firmware, or in more
specialized apparatus constructed to perform the required method steps.

It should be appreciated by those skilled in the art that the specific
embodiments disclosed above may be readily utilized as a basis for
modifying or designing other techniques for carrying out the same
purposes of the present invention. It should also be realized by those
skilled in the art that such equivalent constructions do not depart from
the spirit and scope of the invention as set forth in the appended claims.
CLAIMS

What is claimed is:

1. A method of transmitting data in a communications network having a user connectable to a data center via a channel, comprising the steps of:
   transmitting a primary data stream from the data center to a user, the primary data stream comprising a plurality of data packets conforming to a given protocol and characterized as having periods of inactivity;
   as the primary data stream is being transmitted from the data center to the user, embedding into the primary data stream a secondary broadcast stream comprising a plurality of delay-insensitive data packets conforming to a given protocol, wherein the secondary broadcast stream is embedded into the primary data stream without substantially affecting latency of the data packets comprising the primary data stream; and reconstituting the secondary broadcast stream.

2. The method as described in Claim 1 wherein the secondary broadcast stream is superimposed over the primary data stream by embedding the delay-insensitive data packets during said periods of inactivity in the primary data stream.

3. The method as described in Claim 2 wherein the delay-insensitive data packets are transmitted out-of-sequence relative to each other and out-of-band relative to the primary data stream.

4. The method as described in Claim 2 wherein at least some of the delay-insensitive data packets are transmitted multiple times in the primary data stream.
5. The method as described in Claim 2 wherein at least some of the delay-insensitive data packets are assigned acceptable error-tolerance levels.

6. The method as described in Claim 2 wherein a given delay-insensitive data packet has an optimum size and frequency of retransmission determined by a statistical decomposition of a data-time behavior of the primary data stream.

7. The method as described in Claim 1 wherein the given protocol of the primary data stream is a protocol selected from a group of protocols consisting of: PPP, SLIP and HDLC.

8. The method as described in Claim 1 wherein the given protocol of the secondary broadcast stream excludes a given flag sequence of the primary data stream to thereby control the embedding of the secondary broadcast stream into the primary data stream.

9. The method as described in Claim 8 wherein the embedding and transmission of a given delay-insensitive packet is interrupted upon occurrence of the given flag sequence.

10. The method as described in Claim 1 wherein the given protocol of the secondary broadcast stream includes a start sequence that is restricted from occurring in a header of the primary data stream.
11. The method as described in Claim 10 wherein the primary data stream protocol does not use a flag sequence comprising an opening flag and a closing flag.

12. The method as described in Claim 1 wherein the user is a client machine, the data center is an Internet service provider, and the communication network is the Internet.

13. The method as described in Claim 12 wherein the primary data stream includes information retrieved from a web site connected to the Internet service provider and the secondary broadcast stream includes a stream of advertisements.

14. The method as described in Claim 13 further including the step of displaying the stream of advertisements on a display of the client machine.
15. A transmission system located at a service provider for transmitting data over a set of one or more connections to a set of one or more clients, respectively; the service provider connected to a plurality of servers, comprising:

means responsive to a request from a client for communicating with one of the servers and receiving information to be transmitted to the client;

means responsive to the communicating means for transmitting the information to the client in a primary data stream, the primary data stream comprising a plurality of data packets conforming to a given protocol and characterized as having periods of inactivity; and

means responsive to the transmitting means for embedding into the primary data stream a secondary broadcast stream as the primary data stream is transmitted to the client, the secondary broadcast stream comprising a plurality of delay-insensitive data packets conforming to a given protocol and wherein the secondary stream is superimposed over the primary data stream without substantially affecting latency of the data packets comprising the primary data stream.

16. The system as described in Claim 15 further including means for interrupting the embedding means upon a predetermined occurrence.

17. The system as described in Claim 15 wherein the request is an HTTP request, the server is a web site and the information is a web page.

18. The system as described in Claim 17 wherein the secondary broadcast stream includes a plurality of advertisements.
19. The system as described in Claim 17 wherein the secondary broadcast stream includes content selected from a group of content consisting of: graphics, text, animation, video, audio, programs and combinations thereof.

20. A method of transmitting data in a communications network having at least first and second users each connectable to a data center via a channel, comprising the steps of:

transmitting first and second primary data streams from the data center to the first and second users, respectively, each primary data stream comprising a plurality of data packets conforming to a given protocol and characterized as having periods of inactivity; and

as each primary data stream is being transmitted from the data center to the respective user, embedding into the primary data stream a broadcast stream comprising a plurality of delay-insensitive data packets conforming to a given protocol, wherein the broadcast stream is embedded into each primary data stream without substantially affecting latency of the data packets comprising the respective primary data stream.

21. The method as described in Claim 20 wherein each of the first and second primary data streams differ from each other.

22. The method as described in Claim 20 wherein each of the first and second primary data streams are the same.

23. The method as described in Claim 20 wherein the broadcast stream is embedded into the primary data stream by
embedding the delay-insensitive data packets during said periods of inactivity in the primary data stream.

24. The method as described in Claim 20 wherein the given protocol of the broadcast stream excludes a given flag sequence of the primary data stream.

25. The method as described in Claim 20 wherein the given protocol of the broadcast stream includes a start sequence that is restricted from occurring in a header of the primary data stream.
Figure 1 System Overview

Figure 2: OP Packet

Figure 3: HDLC Packet

Figure 4: Sample HDLC Packet Stream

Figure 5: Sample OP Packet Stream

Figure 6: OP Stream Overlayed on HDLC
Figure 7: Broadcast Stream Being Superimposed on a Single User Stream

Figure 8: Broadcast Stream Superimposed on Multiple User Streams

Figure II: Band Extractor (ADSD)
Figure 1: Overview of Client/User System