Abstract: Without limiting the disclosure, at least one described implementation encapsulates Ethernet data for communication over a cable medium using a TDF structure typical of wireless systems. One encapsulation process receives (1130, 1820) a packet from a first source and a packet from a second source. The two packets have a particular format. The encapsulation process encapsulates (1150, 1850) the two packets into an encapsulated packet having a different format. One extraction process receives (1620, 1920) an encapsulated packet having a given format. The encapsulated packet includes a packet from a first source and a packet from a second source, with the two packets having a format different from the given format. The extraction process extracts (1630, 1930) the two packets from the encapsulated packet. A corresponding signal, as well as apparatuses, for use with the encapsulation and extraction processes are also described.
TECHNICAL FIELD

At least one implementation relates generally to data transmission technology, and more particularly to encapsulating data for transmission over a network.

BACKGROUND

Various communication systems are used for transmission of data. Particular systems encapsulate individual packets as the packets pass from one layer to another layer. The encapsulation allows the layers to provide additional services or features in the communication system.

SUMMARY

According to a general aspect, a packet is received from a first source and a packet is received from a second source. The two packets have a particular format. The two packets are encapsulated into an encapsulated packet having a different format.

According to another general aspect, a signal structured to carry an encapsulated packet having a given format includes two portions. One of the portions is structured to carry a part of the encapsulated packet that includes a packet from a first source. The packet from the first source has a particular format that is different from the given format. Another of the portions is structured to carry a part of the encapsulated packet that includes a packet from a second source. The packet from the second source also has the particular format.

According to another general aspect, an encapsulated packet is received having a given format. The encapsulated packet includes a packet from a first source...
and a packet from a second source, with the two packets having a format different from the given format. The two packets are extracted from the encapsulated packet.

The details of one or more implementations are set forth in the accompanying drawings and the description below. Even if described in one particular manner, it should be clear that implementations may be configured or embodied in various manners. For example, an implementation may be performed as a method, or embodied as an apparatus configured to perform a set of operations or an apparatus storing instructions for performing a set of operations. Other aspects and features will become apparent from the following detailed description considered in conjunction with the accompanying drawings and the claims.

**BRIEF DESCRIPTION OF THE DRAWINGS**

Fig.1 illustrates a simplified exemplary TDF access network architecture.

Fig. 2 illustrates the 802.11 MAC sublayer in OSI reference model.

Fig. 3 illustrates an implementation of a TDF transmission entity in OSI reference model.

Fig. 4 illustrates an implementation of a communication mode entrance procedure.

Fig. 5 illustrates an implementation of a TDF super frame structure.

Fig. 6 illustrates an implementation of a registration procedure.

Fig. 7 illustrates an implementation of an unregistration procedure.

Fig. 8 illustrates an implementation of an alive notification procedure.
Fig. 9 includes a system diagram that depicts an implementation of a TDF network.

Fig. 10 includes a block diagram of an implementation of an AP and a modem from Fig. 9.

Fig. 11 includes a flow diagram of an implementation of an uplink transmission process.

Fig. 12 includes a diagram of an implementation of a one-to-one mapping between an Ethernet packet and a WLAN packet.

Fig. 13 includes a diagram of an implementation of a transformation between multiple Ethernet packets and a single WLAN packet.

Fig. 14 includes a diagram depicting packet flow in the transformation of Fig. 13.

Fig. 15 includes a diagram of an implementation of an EIW header from Fig. 14.

Fig. 16 includes a flow diagram of an implementation of an uplink reception process.

Fig. 17 includes a diagram depicting an implementation for decapsulating packets.

Fig. 18 includes a diagram depicting an implementation of a PADM from Fig. 10.

Fig. 19 includes a flow diagram of an implementation of a downlink transmission process.

Fig. 20 includes a flow diagram of an implementation of a downlink reception process.

DETAILED DESCRIPTION

The discussion of Figs. 1-8 presents a variety of implementations that include one or more novel and inventive aspects or features. At least one of these implementations provides a system that transmits data over a cable using features typical of a wireless system.
In particular, at least one implementation uses time-division multiplexing over a coaxial cable. Such a system allows, for example, a cable television operator to provide television signals in part of a spectrum and to provide additional services over another part of the spectrum. The additional services may include, for example, Internet access, including access for searching the Internet and viewing web pages on the Internet, as well as for receiving services over the Internet (such as, for example, video-on-demand).

The discussion of Figs. 9-19 presents additional implementations, and at least one of the additional implementations extends the discussion of Figs. 1-8 by describing novel and inventive uses of encapsulation. One particular implementation includes a modem that receives Ethernet packets from multiple hosts. Each host may be attempting to communicate with a different web site, through a router. The modem encapsulates these packets into a single packet that is formatted according to a format structure, or protocol, for wireless transmission. The encapsulated packet, however, is sent over a coaxial cable for receipt by the router. The router in turn sends the packets, in one implementation, to the different web sites with which each of the hosts is attempting to communicate.

The encapsulation used by the above-described implementation provides an increase in throughput, as compared to a system that encapsulates only one packet at a time. Thus, the overhead of the wireless format structure is spread out over multiple Ethernet packets. This stands in contrast to the conventional use of encapsulation which, for example, allows additional features to be provided by another communication layer,
or ensures backward compatibility by preserving a legacy frame structure in the encapsulated data. Further, the encapsulation of the above-described implementation also allows, depending on system design, for data from multiple sources to be encapsulated together, and for data intended for different end-users (for example, different web sites, or different hosts) to be encapsulated together.

This application now provides a description of Figs. 1-8. Note that headers are used for various sections of the description of Figs. 1-8. The header of a given section is not to be construed as limiting the disclosure of that section to the topic of the header, nor as limiting the disclosure of other sections to topics other than that of the header. The headers are exemplary, and are intended as a general aid to the reader. The headers are not intended to constrain the flow of the disclosure nor to restrict the applicability or generality of the disclosure.

General description

Application scenario

In order to provide data service over existing coaxial cable TV system (CATV), at least one implementation deploys a time divisional function (TDF) protocol compliant Access Point (AP) and stations (STAs) in the cable access network. The AP and STAs are connected via splitters in the hierarchical tree structure. In this way, the user at home can access the remote IP core network via the cable access network. The detailed network topology is illustrated as illustrated in Fig.1.
As can be seen from Fig.1, in this typical access network infrastructure, there is a TDF protocol compliant AP which has one Ethernet Interface in connection with the IP core network, and one coaxial cable interface in connection with the cable access network. On the other end of the cable access network, there are TDF protocol compliant STAs, i.e. terminals, which connect with the cable access network via the coaxial cable interface and connect with the home LAN (Local Area Network) via the Ethernet interface.

According to at least one implementation, both TDF APs and STAs implement the protocol stack separately in logically link control sublayer, MAC sublayer and physical layer, according to 802.11 series specifications. However, in the MAC sublayer, the TDP APs and STAs replace the 802.11 frame transmission entity with TDF frame transmission entity. So, the MAC sublayer for TDF APs and STAs is composed of 802.11 frame encapsulation/decapsulation entity and TDF frame transmission entity, while MAC sublayer for 802.11 compliant APs and STAs consists of 802.11 frame encapsulation/decapsulation entity and 802.11 frame transmission entity. For an integrated AP and STA, the TDF frame transmission entity and 802.11 frame transmission entity may co-exist at the same time, to provide both 802.11 and TDF functionality. The switch between the two modes can be realized by manually or dynamically configuration.

**Basic approach**

The main idea of the TDF protocol is to transmit IEEE802.11 frames in the coaxial cable media instead of over the air. The purpose of utilizing the IEEE802.11
mechanism is to make use of the mature hardware and software implementation of 802.11 protocol stacks.

The main feature of TDF is its unique medium access control method for transmitting IEEE802.11 data frames. That is, it doesn't utilize the conventional IEEE802.11 DCF (Distributed Coordination Function) or PCF (Point Coordination Function) mechanism to exchange MAC frames, which include MSDU (MAC Service Data Unit) and MMPDU (MAC Management Protocol Data Unit). Instead, it uses time division access method to transmit MAC frames. So the TDF is an access method which defines a detailed implementation of frames transmission entity located in MAC sublayer.

For the purpose of comparison, here we illustrate IEEE802.11 MAC sublayer protocol in the OSI reference model as shown in the Fig.2. While the exact location for TDF protocol in the OSI reference model is illustrated in the Fig.3.

Communication mode entrance procedure

Currently, there are two communication modes proposed for the TDF compliant stations described as below. One is the standard IEEE802.11 operation mode, which obeys to the frame structure and transmission mechanism defined in IEEE802.11 series standard; the other is in TDF operation mode, the detailed information about which will be discussed in the following paragraphs.

The strategy of determining entering into which operation mode when a TDF STA is started is indicated in the Fig.4. Once a TDF STA receives a synchronization frame from an AP, it is enabled to entering into TDF mode, if there is
no synchronization frame received within a preset timeout, then the TDF STA remains or shifts into IEEE802.11 mode.

TDF protocol functional description

Access method

The physical layer in a TDF station may have multiple data transfer rate capabilities that allow implementations to perform dynamic rate switching with the objective of improving performance and device maintenance. Currently, TDF station may support three types of data rates: 54Mbps, 18Mbps and $\beta$Mbps. The data service is provided mainly in 54Mbps data rate. When there are some problems for a station to support 54Mbps data transmission, it may temporarily switch to 18Mbps data rate. The $\beta$Mbps data rate operation mode is designed for the purpose of network maintenance and station debugging.

The data rate may be configured statically before a TDF station enters the TDF communication procedure, and remain the same during the whole communication process. On the other hand, the TDF station may also support dynamical data rate switch during the service. The criteria for the data rates switch may be based on the channel signal quality and other factors.

The fundamental access method of TDF protocol is Time Division Multiple Access (TDMA), which allows multiple users to share the same channel by dividing it into different time slots. The TDF STAs transmit in rapid succession for uplink traffic, one after the other, each using their own time slot in a TDF super frame assigned by the TDF AP. For downlink traffic, the STAs share the channels, and select the data or management frames
targeting to them by comparing the destination address information in the frames with their address. Fig. 5 illustrates an example of TDF super frame structure and the time slots allocation for a typical TDF super frame when there are m STAs which simultaneously compete for the uplink transmission opportunity.

As shown in Fig. 5, there are fixed tdftotalTimeSlotNumber timeslots per TDF super frame, which is composed of one synchronization time slot used to send clock synchronization information from TDF AP to TDF STAs; one contention time slot used to send registration request for uplink time slot allocation; tdfulinkTimeSlotNumber uplink time slots used by the registered TDF STAs to send data and some management frames to TDF AP one after another; and tdffdownlinkTimeSlotNumber downlink time slots used by TDF AP to transmit data and registration response management frames to the modems. Except the synchronization time slot, all other time slots, which are named as common time slots, have same duration whose length equals with tdffCommonTimeSlotDuration. The value of tdffCommonTimeSlotDuration is defined to allow the transmission of at least one largest IEEE802.11 PLCP (physical layer convergence protocol) protocol data unit (PPDU) in one normal time slot for the highest data rate mode. The duration of synchronization time slot, tdffSyncTimeSlotDuration, is shorter than that of the common time slot, because the clock synchronization frame, which is transmitted from TDF AP to TDF STA in this time slot, is shorter than the 802.11 data frame.

As a result, the duration of one TDF super frame, defined as tdffSuperframeDuration, can be calculated by:
tdfSuperframeDuration = tdfSyncTimeSlotDuration +
tdfCoitimonTimeSlotDuration * \( (tdfTotalTimeSlotNumber - 1) \)

The relationship between tdfTotalTimeSlotNumber, tdfUplinkTimeSlotNumber and tdfDownlinkTimeSlotNumber satisfies the following equality:

\[
tdfTotalTimeSlotNumber = tdfUplinkTimeSlotNumber + tdfDownlinkTimeSlotNumber + 2
\]

Furthermore, the number of allocated uplink time slots for the TDF STAs in a TDF super frame may change from one to tdfUplinkTimeSlotThreshold. Accordingly, the available downlink time slots in a TDF super frame may change from \( (tdfTotalTimeSlotNumber-2) \) to \( (tdfTotalTimeSlotNumber-2-tdfMaximumUplinkTimeSlotNumber) \).

Every time when there is one TDF STA which asks for an uplink time slot, the TDF AP will deduce one or more time slots from the available downlink time slots, and then allocate these time slots to the TDF STA, as long as the uplink time slots number won't exceed tdfMaximumUplinkTimeSlotNumber after that. The value of tdfMaximumUplinkTimeSlotNumber may vary in different implementations. But it must be carefully chosen so that there is at least one downlink time slot available for an associated TDF STA in order to guarantee the QoS of data service. Furthermore, all successive time slots that will be used by the same TDF STA or AP for same direction transmission can be merged to send MAC frames continuously to avoid the wastes at the edge of these time slots caused by the unnecessary conversion and guarding.

In current implementation, the tdfCommonTimeSlotDuration is about 300us, which is enough
for the TDF STA to transmit at least one largest 802.11 PPDU in one common time slot for 54M mode, and there are total 62 time slots per TDF super frame. In these time slots, there are 20 uplink time slots and 40 downlink time slots in this way. When there are 20 STAs, each TDF STA can be guaranteed that it has access to 680kbps uplink data rate and shares 30Mbps (40 continuous time slots) downlink data rate; when there are 30 STAs, each TDF STA can be guaranteed that it has access to 680kbps uplink data rate and shares 22.5Mbps (30 continuous time slots) downlink data rate. The `tdfMaximumUplinkTimeSlotNumber` is 30. Finally, the value of `tdfSuperframeDuration`, which is the total duration of 61 common time slots and one synchronization time slot, is about 18.6ms and it can be defined to different value for different usage. For example, if there is only 1 TDF STA, it can be guaranteed that it has 4 time slots to achieve about 18Mbps uplink data rate and own 18Mbps (4 continuous time slots) downlink data rate. In this way, the value of `tdfSuperframeDuration`, which is the total duration of nine data timeslots and one synchronization timeslot, is about 4ms.

Frame formats

In the 802.11 specification, three major frame types exist. Data frames are used to exchange data from station to station. Several different kinds of data frames can occur, depending on the network. Control frames are used in conjunction with data frames to perform area clearing operations, channel acquisition and carrier-sensing maintenance functions, and positive acknowledgement of received data. Control and data frames work in conjunction to deliver data reliably from station to station. More specifically, one important feature for
the data frames exchanging is that there is an acknowledgement mechanism, and accordingly an Acknowledgement (ACK) frame for every downlink unicast frame, in order to reduce the possibility of data loss caused by the unreliable wireless channel. Finally, management frames perform supervisory functions: they are used to join and leave wireless networks and move associations from access point to access point.

However, in TDF system, because TDF STAs passively waits for the Synchronization frame from TDF AP to find the targeting TDF AP, there is no need for the classical Probe Request and Probe Response frames. Furthermore, the frames are exchanged in coaxial cable instead of in the air, so it isn't necessary to define RTS and CTS frames to clear an area and prevent the hidden node problem, and to define ACKs frames to ensure the reliability of delivery of data frames.

So, in TDF protocol, we only use some useful 802.11 MSDU and MMPDU types for data over coaxial cable scenario. For example, we utilize the data subtype in data frame types, which is used to encapsulate the upper layer data and transmit it from one station to another. Furthermore, to cope with clock synchronization requirement in TDF system, we define a new kind of management frame—Synchronization frame; and to realize the functionality of uplink time slot request, allocation and release, we defines other four kinds of management frames that are Registration request, Registration response, Unregistration request and Alive notification.

To summarize it, we have defined four new subtypes in management frame type in TDF protocol. The following...
Table 1 defines the valid combinations of type and subtype added in TDF protocol. Table 1 shows valid type and subtype for TDF frames added in TDF protocol.

<table>
<thead>
<tr>
<th>Type description</th>
<th>Subtype description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>Synchronization</td>
</tr>
<tr>
<td>Management</td>
<td>Registration request</td>
</tr>
<tr>
<td>Management</td>
<td>Registration response</td>
</tr>
<tr>
<td>Management</td>
<td>Unregistration request</td>
</tr>
<tr>
<td>Management</td>
<td>Alive notification</td>
</tr>
</tbody>
</table>

TDF access procedure

TDF AP finding and clock synchronization procedure

TDF protocol depends a great deal on the distribution of timing information to all the nodes. Firstly, the TDF STA listens to a Synchronization frame to decide if there is a TDF AP available. Once it enters the TDF communication procedure, it uses the Synchronization frame to adapt the local timer, based on which the TDF STA shall decide if it is its turn to send the uplink frames. At anytime, TDF AP is master and TDF STA is slave in synchronization procedure. Further, if it hasn't received any Synchronization frame from the associated AP for a predefined threshold period, which is defined as tdfSynchronizationCycle, the TDF STA will think that the AP has quit the service, and then it will stop the TDF communication process and start to look for any TDF AP by listening to the Synchronization frame again.

In the TDF system, all STAs associated with the same TDF AP shall be synchronized to a common clock. The TDF AP shall periodically transmit special frames called
Synchronization that contains its clock information to synchronize the modems in its local network. Every TDF STA shall maintain a local timing synchronization function (TSF) timers, to ensure it is synchronized with the associated TDF AP. After receiving a Synchronization frame, a TDF STA shall always accept the timing information in the frame. If its TSF timer is different from the timestamp in the received Synchronization frame, the receiving TDF STA shall set its local timer according to the received timestamp value. Further, it may add a small offset to the received timing value to account for local processing by the transceiver.

Synchronization frames shall be generated for transmission by the TDF AP once every TDF super frame time units and sent in the Sync time slot of every TDF super frame.

Registration procedure

Fig. 6 illustratively describes the whole procedure of registration. Once a TDF STA has acquired timer synchronization information from the Synchronization frame, it will learn when time slot 0 starts. If a TDF STA doesn't associate with any TDF AP, it will try to register with the specific TDF AP, which sent the Synchronization frame, by sending Registration request frames to TDF AP during the contention time slot, which is the second time slot in a TDF super frame. The duration of contention time slot, which equals with tdfCommonTimeSlotDuration, and the Registration request frame structure should be carefully designed to allow for sending at least tdfMaximumUplinkTimeSlotNumber Registration request frames in one contention time slot. Based on the design, the contention time slot is divided
As soon as it finds the targeting TDF AP, a TDF STA will choose one sub-timeslot in the contention time slot to send Registration request frame to the TDF AP according to the following method:

A. Every time when it is allocated an uplink time slot, a TDF STA will store the allocated uplink time slot number, defined as tdfAllocatedUplinkTimeSlot, which indicates the time slots' location in the whole uplink time slots pool and ranges from 1 to tdfMaximumUplinkTimeSlotNumber.

B. The TDF AP should try its best to allocate same uplink time slot to the same TDF STA every time when it asks for an uplink time slot.

C. When it is time to decide to choose which sub-timeslot to send Registration request frame, if there is a stored tdfAllocatedUplinkTimeSlot value, the TDF STA will set the sub-timeslot number as same as tdfAllocatedUplinkTimeSlot; if there isn't such a value, the TDF STA will randomly choose one sub-timeslot in the tdfMaximumUplinkTimeSlotNumber available sub-timeslots. It will send the Registration request frame to the TDF AP in the randomly chosen sub-timeslot.

The purpose for this kind of operation is to reduce the chance of collision when there are many STAs start at the same time and try to register with the same TDF AP simultaneously.

The TDF STA will list all data rates it supports at that time and also carry some useful information such as the received signal Carrier/Noise ratio in the
Registration request frame. It may send several successive Registration request frames with different supported data rates, starting from the highest data rate. After sending out the frame, the TDF STA will listen for the Registration response frames from the TDF AP.

After receiving a Registration request frame from a TDF STA, based on the following method, the TDF AP will send different kinds of Registration response frames back to the TDF STA in the downlink time slots:

A. If the already allocated uplink time slots equals with tdfMaximumUplinkTimeSlotNumber, the TDF AP will put an uplinkTimeSlotUnavailable indicator in the frame body.

B. If the TDF AP doesn't support any data rates listed in the supportedDataratesSet in the Registration request management frame, the TDF AP will put an unsupportedDatarates indicator in the frame body.

C. If there are uplink timeslots available to allocate and common data rates that both the TDF AP and TDF STA can support, the AP will allocate one uplink time slot and choose a suitable common data rates according to some information such as Carrier/Noise ratio in the STA's Registration request frame, and then send a Registration response frame to the TDF STA. In the frame body, the information about the allocated uplink time slot and the chosen data rate will be contained.

After a successful registration process, the TDF STA and TDF AP will reach an agreement on which uplink time slot and data rate to use.
Fragmentation/defragmentation procedure

In TDF protocol, the time slot duration for the transmission of MSDU is fixed as tdfCommonTimeSlotDuration. In some data rates, when the MSDU's length is more than a threshold, it is impossible to transmit it in a single time slot. So when a data frame for uplink transmission is longer than the threshold, which is defined as tdfFragmentationThreshold and varies depending on different data rates, it shall be fragmented before scheduled for transmitting. The length of a fragment frame shall be an equal number of octets (tdfFragmentationThreshold octets), for all fragments except the last, which may be smaller. After fragmentation, the fragmented frames shall be put into the outgoing queue for transmission to the TDF AP. This fragmentation procedure may run in the TDF frame transmission entity or in the upper layer by using the tdfFragmentationThreshold dynamically set in the TDF frame transmission entity.

At the TDF AP end, each fragment received contains information to allow the complete frame to be reassembled from its constituent fragments. The header of each fragment contains the following information that is used by the TDF AP to reassemble the frame:

A. Frame type
B. Address of the sender, obtained from the Address 2 field
C. Destination address
D. Sequence Control field: This field allows the TDF AP to check that all incoming fragments belong to the same MSDU, and the sequence in which the fragments should be reassembled. The sequence number within the Sequence Control field remains the same for all fragments of a
MSDU, while the fragment number within the Sequence Control field increments for each fragment.

E. More Fragments indicator: Indicates to TDF AP that this is not the last fragment of the data frame.

Only the last or sole fragment of the MSDU shall have this bit set to zero. All other fragments of the MSDU shall have this bit set to one.

The TDF AP shall reconstruct the MSDU by combining the fragments in order of fragment number subfield of the Sequence Control field. If the fragment with the More Fragments bit set to zero has not yet been received, the TDF AP will know that the frame is not yet complete. As soon as the TDF AP receives the fragment with the More Fragments bit set to zero, it knows that no more fragments may be received for the frame.

The TDF AP shall maintain a Receive Timer for each frame being received. There is also an attribute, tdfMaxReceiveLif etime, which specifies the maximum amount of time allowed to receive a frame. The receive timer starts on the reception of the first fragment of the MSDU. If the receive frame timer exceeds tdfMaxReceiveLif etime, then all received fragments of this MSDU are discarded by the TDF AP. If additional fragments of a directed MSDU are received after its tdfMaxReceiveLif etime is exceeded, those fragments shall be discarded.

Uplink transmission procedure

After receiving the Registration response frame from the TDF AP, the TDF STA will analyze the frame body to see if it is granted an uplink time slot. If not, it will stop for a while and apply for the uplink time slot later. If yes, it will start to transmit uplink traffic
during the assigned time slot using the data rate indicated in the Registration response frame.

At the beginning of the uplink transmission during the assigned timeslot, the TDF STA will send the first frame in its outgoing queue to the TDF AP if there is at least one outgoing frame in the queue. After that, the TDF STA will check the second uplink frame's length and evaluate if it is possible to send it during the remaining duration in the assigned timeslot. If not, it will stop the uplink transmission procedure and wait for sending it in the assigned timeslot during the next TDF super frame. If yes, it will immediately send the second frame to the destination TDF AP. The sending procedure will continue to run in this way until the assigned timeslot has ended, or there isn't any uplink frame to transmit.

**Downlink transmission procedure**

In the whole TDF communication procedure, the total downlink time slots number may change dynamically due to the changing associated STAs number. When the TDF AP prepares to send frames to the associated STAs, it will compare the time left in the remaining downlink time slots with the duration needed for transmitting the specific downlink frame using the agreed data rate. Then based on the result, it will decide if the frame should be transmitted with the specific data rate during this TDF super frame. Furthermore, TDF AP doesn't need to fragment any downlink frames.

When it isn't time for the associated STA to send uplink traffic, the STA will always listen to the channel for the possible downlink frames targeting to it.
Unregistration procedure

As shown in Fig. 7, if the TDF STA decides to quit the TDF communication procedure, it shall send an Unregistration request frame to the associated TDF AP during its uplink time slot, in order to inform the TDF AP to release the allocated uplink time slot resource for it. After receiving the Unregistration request frame, the TDF AP will free the uplink time slot assigned for the TDF STA and put it into free time slots pool for the future use.

Alive notification procedure

Now with reference to Fig. 8, to release the resources as soon as possible when a TDF STA unexpectedly crashes or shuts down, the TDF STA must report its aliveness by sending an Alive notification frame periodically to TDF AP during its uplink time slot period. If there isn't any Alive notification frame for a predefined threshold period which is named as tdfAliveNotificationCycle, the associated TDF AP will think that the TDF STA has quit the service, and then release the uplink time slot allocated for the TDF STA, just like receiving an Unregistration request frame from the TDF STA.

In order to ensure coexistence and interoperability on multirate-capable TDF STAs, this specification defines a set of rules that shall be followed by all stations:

A. The Synchronization frames shall be transmitted at the lowest rate in the TDF basic rate set so that they will be understood by all STAs.

B. All frames with destination unicast address shall be sent on the supported data rate selected by the registration mechanism. No station shall transmit a
unicast frame at a rate that is not supported by the receiver station.

c. All frames with destination multicast address shall be transmitted at the highest rate in the TDF basic rate set.

The description of Figs. 9-19 follows. Figs. 9-19 describe implementations that may be used, for example, with one or more systems described by Figs. 1-8. Of course, the features and aspects of the implementations of Figs. 9-19 may be used with other systems.

As described above, a TDF protocol can replace the conventional 802.11 DCF (distributed coordination function) or PCF (point coordination function) mechanism. Such a system can take advantage of a wide deployment of WLAN (802.11) network, and a WLAN chipset that may be getting more and more mature and cheap. This system provides a cost effective solution for bidirectional communication of CATV network by transmitting WLAN signals in cable network, even though the WLAN protocols are created for transmission/reception in an air environment rather than a cable network. In this system, the fundamental access method of TDF protocol is TDMA, which allows multiple users to share the same channel by dividing it into different time slots. The TDF stations transmit in rapid succession for uplink traffic, one after the other, each using their own time slot in a TDF superframe assigned by the TDF AP (access point). For downlink traffic, the stations share the channels (as shown, for example, in the TDF superframe of Fig. 5), and select the frames targeting them by comparing the destination address information in the frames with their address.
Referring to Fig. 9, a typical TDF network 900 is shown. The network 900 provides a connection from user homes 910 and 920 to the Internet (or another resource or network) 930. The user homes 910 and 920 connect through an access point (AP) 940 over a cable system 950. The AP 940 may be located, for example, in a neighbourhood of the homes 910 and 920, or in an apartment building that includes the homes (apartments, in this case) 910 and 920. The AP 940 may be owned by a cable operator, for example. The AP 940 is further coupled to a router 960 over an Ethernet network 970. The router 960 is also coupled to the Internet 930.

As should be clear, the term "coupled" refers to both direct connections (no intervening components or units) and indirect connections (one or more intervening components and/or units). Such connections may be, for example, wired or wireless, and permanent or transient.

The user homes 910 and 920 may have a variety of different configurations, and each home may be differently configured. As shown in the network 900, however, the user homes 910 and 920 each include a station (referred to as a modem) 912 and 922, respectively. The modems 912, 922 are coupled to a first host (host1) 914, 924, and a second host (host2) 916, 926, over an Ethernet network 918, 928, respectively. Each host 914, 916, 924, and 926 may be, for example, a computer or other processing device or communication device.

There are various ways in which the network 900 may allow multiple hosts (for example, 914, 916, 924, and 926) to connect to the router 960. Four implementations are discussed below, considering only the modem 912 and hosts 914 and 916, for simplicity.
In a first method, the modem 912 acts as another router. The hosts 914 and 916 are identified by their IP addresses, and the modem 912 routes IP packets from the hosts 914 and 916 to the router 960. This method typically requires the modem 912 to run router software, which requires additional memory and increased processing power.

In a second method, the modem 912 acts as a bridge. The modem 912 and the AP 940 use the standard wireless distribution system (WDS) mechanism to convey layer 2 packets to the router 960. The hosts 914 and 916 are identified by their media access control (MAC) addresses. This method is part of the 802.11 standard and can serve multiple hosts simultaneously. However, not all APs and modems support WDS, and those that do support WDS often have only limited support. For example, with some APs and modems, you cannot use Wi-Fi protected access (WPA) with WDS, and this may introduce security problems.

In a third method, the modem 912 uses MAC masquerade to change the source MAC address of Ethernet packets (the source being one of the hosts 914 and 916) to its own MAC address. So from the point of view of the router 960, the router 960 only sees the modem 912. The modem 912 can only serve one host at one time with this method.

In a further method, the modem 912 uses encapsulation, as described in further detail below. Each of the above methods has advantages and disadvantages, and these advantages and disadvantages may vary depending on the implementation. However, the encapsulation method provides particular advantages in that it generally allows the modem to be simpler by not requiring that the modem run router software, it does not typically introduce security problems, and it can serve multiple hosts at one time.
Additionally, the encapsulation method avoids the large overhead associated with the first three methods, which transfer each packet from a host by using a single WLAN packet. Thus, the first three methods incur the overhead of the WLAN packet for every packet transferred from a host, and the throughput is correspondingly reduced. Such inefficiency is typically aggravated in the TDF environment. In the TDF environment, the duration of the time slot is fixed, and the time slot is designed to allow only one WLAN packet to transmit in one slot. Thus, only one host packet can be transmitted in each time slot.

Accordingly, the encapsulation method generally provides one or more of a variety of advantages. Such advantages include, for example, simpler router design and operation, increased security, serving multiple hosts, and increased efficiency and throughput.

In summary, at least one implementation of the encapsulation method includes encapsulating multiple Ethernet packets into one WLAN packet. The WLAN packet will be as big as the maximum length allowed by the TDF time slot. The AP (for example, another modem) will decapsulate the WLAN packet into individual Ethernet packets and send them to the Router. For communication in the reverse direction, a modem will decapsulate a WLAN packet and send the individual Ethernet packets to the host(s).

Referring to Fig. 10, an illustration 1000 includes multiple modems, two of which are explicitly shown, and an AP. The illustration includes a modem #1 1010, a modem #N 1020, and an AP 1030, with each of the modems 1010 and 1020 coupled to the AP 1030 over a cable network 1040. Other implementations use separate cable networks for each of the modems.
The modems 1010 and 1020, and the AP 1030 include functional components of the same name, although some of the external connections are different and the components themselves perform different functions for a modem and for an AP. Thus, a common unit is provided that serves as both a modem and an AP. However, it should be clear that different units could be designed for a modem and for an AP, with the different units performing only those functions required of a modem or an AP, respectively.

The modem 1010 includes a local applications layer 1011, followed by a TCP/IP layer 1012, followed by a bridge 1014. The bridge 1014 is coupled to an Ethernet interface 1015, a packet aggregation/deaggregation module (PADM) 1016, and a WLAN interface 1017. The PADM 1016 is also coupled to the WLAN interface 1017. The Ethernet interface 1015 is coupled to an Ethernet network 1052, that is coupled to a first host (host1) 1054 and a second host (host2) 1056.

The modem 1020 is analogous to the modem 1010. However, the modem 1020 is coupled to an Ethernet network 1062, and the Ethernet network 1062 is coupled to a first host (host1) 1064 and a second host (host2) 1066. The components of the modem 1020 are shown as being identical to those of the modem 1010. However, it should be clear that various configuration parameters, for example, will be different when the modems 1010 and 1020 are set up and operational.

The AP 1030 includes a local applications layer 1071, followed by a TCP/IP layer 1072, followed by a bridge 1074. The bridge 1074 is coupled to an Ethernet interface 1077, a PADM 1076, and a WLAN interface 1075. The PADM 1076 is also coupled to the WLAN interface 1075. The Ethernet interface 1077 is coupled to an Ethernet network 1082, which in turn is coupled to a router 1090.
The WLAN interfaces 1017 and 1075 are communicatively coupled to each other over the cable network 1040.

The router 1090 is further coupled to the Internet 1095. Thus, a connection exists between the hosts 1054, 1056, 1064, 1066, and the Internet 1095.

The various local application layers (1011, 1071) are standard layers for running local applications and interfacing with other layers in the architecture. The various TCP/IP layers (1012, 1072) are standard layers for running TCP/IP and for providing the services typically provided by such layers, including interfacing to other layers in the architecture. The various Ethernet interfaces (1015, 1077) are standard units for interfacing to/from an Ethernet network. Such interfaces 1015, 1077 transmit and receive Ethernet packets and operate according to the Ethernet protocol.

The various WLAN interfaces (1017, 1075) are units for interfacing to/from a WLAN network. Such interfaces 1017, 1075 transmit and receive WLAN packets and operate according to the WLAN protocol. However, the WLAN interfaces 1017, 1075 are actually coupled, in the illustration 1000, to a cable network 1040 rather than using wireless communication.

The Ethernet and WLAN interfaces 1015, 1017, 1075, and 1077 may be implemented, for example, in hardware such as a plug-in card for a computer. The interfaces may also largely be implemented in software such as a program that performs the functions of the interface using instructions that are implemented by a processing device. Such an interface will generally include a portion for receiving the actual signal (for example, a connector) and for buffering the received signal (for example, a transmit/receive buffer), and typically a
portion for processing the signal (for example, all or part of a signal processing chip).

The various bridges (1014, 1074) are units that forward packets between an Ethernet interface and a WLAN interface. A bridge may be software or hardware implemented, or may only be a logical entity. Standard implementations for a bridge include a processing device (such as an integrated circuit) or a set of instructions running on a processing device (such as a processor running bridge software).

The PADMs 1016 and 1076 perform a variety of functions, including packet encapsulation and decapsulation, which are further described below. The PADMs 1016 and 1076 may be implemented in, for example, software, hardware, firmware, or some combination. Software implementations include, for example, a set of instructions such as a program for running on a processing device. Hardware implementations include, for example, a dedicated chip such as an application specific IC (ASIC).

Referring to Fig. 11, a process 1100 depicts a process for transferring packets from a host to a modem. The packets are further transmitted from the modem for receipt by an AP, and for eventual delivery to a router and then a final destination. This process 1100 is also referred to as an uplink transmission process.

The process 1100 includes the modem connecting to the AP (1110) using, for example, processes described earlier in this application. Such processes may include, for example, standard WLAN protocols including authentication and association operations.

The process 1100 then includes one or more hosts sending one or more packets (1120) to the modem, and the modem receiving the sent packet(s) (1130). Note that the
send packets are for receipt by a router which will deliver the packet(s) to the final destination(s). In the implementation of Fig. 10, the modem 1010 receives the sent packets, from one or more of the hosts 1054 and 1056, over the Ethernet network 1052 through the Ethernet interface 1015.

The modem then determines that the packet(s) is (are) to be sent over a WLAN interface (1140). The modem makes this determination (1140) by recognizing that the router is accessed over the WLAN interface, as opposed, for example, to being accessed over another interface (not shown). In the implementation of Fig. 10, the modem 1010 sends the received packet(s) to the bridge 1014, and the bridge 1014 makes this determination (1140).

The modem then encapsulates multiple packets for the router, including the one or more received packets (1150). The encapsulation (1150) may include packets received from multiple hosts, for example, from hosts 1054 and 1056 in the implementation of Fig. 10. Further, the encapsulation may include the packet(s) received in operation 1130 and packets received earlier and stored in a queue.

In an implementation that does not encapsulate multiple packets, the implementation may use a bridge to map Ethernet packets to individual WLAN packets, encapsulating each Ethernet packet individually. Such encapsulation may, for example, include the entire Ethernet packet as a data portion of a WLAN packet and add an additional WLAN header.

Further, implementations that do not encapsulate multiple packets need not even encapsulate the individual Ethernet packets. Rather, such implementations may transform individual Ethernet packets into individual WLAN packets by, for example, replacing the Ethernet
header with a WLAN header and by optionally adding one or more additional fields.

For example, referring to Fig. 12, a transformation 1200 is shown that receives an Ethernet packet 1210 including an Ethernet header 1220 and a data portion 1230. The transformation 1200 produces a WLAN packet 1240 that includes a WLAN header 1250, the data portion 1230, and a frame check sequence (FCS) 1260.

However, implementing operation 1150 includes encapsulating multiple Ethernet packets into a single WLAN packet. One implementation of operation 1150 is illustrated in Fig. 13.

Referring to Fig. 13, a transformation 1300 receives multiple Ethernet packets, including Ethernet packets 1310, 1312, and 1314, and produces a single WLAN packet 1318. The Ethernet packets 1310, 1312, and 1314 each include an Ethernet header 1320, 1322, and 1324, respectively, and a data portion 1326, 1328, and 1329, respectively.

The Ethernet packets 1310, 1312, and 1314 may originate from the same host, or from different hosts. Further, although the Ethernet packets 1310, 1312, and 1314 are being encapsulated for sending to a router, the final destinations of the Ethernet packets 1310, 1312, and 1314 may be different. For example, each of the Ethernet packets 1310, 1312, and 1314 may be destined for different Internet sites with which the one or more hosts are communicating (or attempting to communicate).

The transformation 1300 is shown as including two intermediate operations. However, other implementations do not perform any intermediate operations, and still other implementations perform more intermediate operations.
The first intermediate operation is transforming the Ethernet packets into extended Ethernet packets. The Ethernet packets 1310, 1312, and 1314 are transformed into extended Ethernet packets 1330, 1332, and 1334, respectively. In the transformation 1300, the entire Ethernet packets 1310, 1312, and 1314 are included as data portions 1336, 1338, and 1340, respectively, of the extended Ethernet packets 1330, 1332, and 1334. The extended Ethernet packets 1330, 1332, and 1334 also include, respectively, optional headers 1342, 1343, and 1344, as well as optional tails 1346, 1347, and 1348. The headers 1342, 1343, and 1344, and the tails 1346, 1347, and 1348 may include a variety of different pieces of information, whether typical for headers/tails or not, such as, for example, packets numbers, acknowledgment and retransmission information, addresses for sources and/or destinations, and error checking information.

The second intermediate operation includes transforming the extended Ethernet packets into a single "Ethernet in WLAN" (EIW) packet 1350. The EIW packet 1350 includes data portions for each of the extended Ethernet packets. Two possible transformations are shown. The first is illustrated by solid arrows 1370 and the second is illustrated by dashed arrows 1375.

As shown by the solid arrows 1370 in the transformation 1300, data portions 1352, 1353, and 1354 correspond, respectively, to the included extended Ethernet packets 1330, 1332, and 1334. The EIW packet 1350 further includes an optional header 1356 (also referred to as an EIW header) and an optional tail 1358, which may include, for example, any of the information previously described for headers/tails.

If no header or tail is inserted into an extended Ethernet packet, then the data portion of the extended...
Ethernet packet (for example, the data portion 1336) becomes the data portion of the EIW packet (for example, the data portion 1352). Further, even if a header or tail is inserted into the extended Ethernet packet, an implementation may discard/ignore the header or tail when forming the EIW packet. In either of these cases, the data portions of the extended Ethernet packet and the EIW packet have the same data.

As shown by the dashed arrows 1375 in the transformation 1300, the data portions 1352, 1353, and 1354 need not correspond, respectively, to the extended Ethernet packets 1330, 1332, and 1334. That is, a data portion of an EIW packet need not contain an entire extended Ethernet packet. As indicated by the dashed arrows 1375, an extended Ethernet packet may be divided into the data portions of two EIW packets.

More specifically, the implementation illustrated by the dashed arrows 1375 shows that (1) a second part of the extended Ethernet packet 1330 is put into the data portion 1352 of the EIW packet 1350, (2) the entire extended Ethernet packet 1332 is put into the data portion 1353 of the EIW packet 1350, and (3) a first part of the extended Ethernet packet 1334 is put into the data portion 1354 of the EIW packet 1350. Thus, in one scenario for the EIW packet 1350, (1) the first data portion 1352 contains a partial extended Ethernet packet, and (2) the last data portion 1354 contains a partial extended Ethernet packet, while (3) the middle data portions (1353 and any other data portions not explicitly shown) contain complete extended Ethernet packets.

Although not shown, it should be clear that the first part of the extended Ethernet packet 1330 may be placed in a data portion of a previous EIW packet, and (2) a
second part of the extended Ethernet packet 1334 may be placed in a data portion of a subsequent EIW packet.

In the final stage of the transformation 1300, the EIW packet 1350 is included as a data portion 1360 in the WLAN packet 1318. The WLAN packet 1318 also includes a WLAN MAC header 1362 and an FCS 1364.

As should be clear, not all implementations use all of the optional headers and tails, nor even use all (or any) of the optional intermediate operations (also referred to as stages). For example, other implementations only copy part of the extended Ethernet packets into the EIW packet, in order to fit more original data (for example, data portions 1326, 1328, and 1329) into a fixed-duration timeslot. As should be clear, the determination of which headers and tails are used, as well as how many intermediate operations are included, may vary for each implementation based on design goals and constraints.

Referring to Fig. 14, a diagram 1400 shows how one implementation of a PADM encapsulates Ethernet packets. The PADM maintains an ingress queue 1410 into which each incoming Ethernet packet is placed. The PADM concatenates the Ethernet packets into a string 1420, and adds an EIW header 1430 and a WLAN header 1440. Depending on the information included in the headers 1430 and 1440, these headers 1430 and 1440 may be constructed ahead of time or after concatenating the Ethernet packets. For example, at least one implementation includes in the EIW header 1430 a number representing the number of Ethernet packets in the string 1420. Given that Ethernet packets may have a variable length, this number is not typically available until after the Ethernet packets have been assembled into the string 1420. As should be clear,
the headers 1430 and 1440 may be defined to accommodate the needs of a particular implementation.

Referring to Fig. 15, a format 1500 of one implementation of an EIW header is shown. The format 1500 includes a field 1510 for sequence and ack numbers, a total packet number 1520, and a series of packet descriptors, including one descriptor for each Ethernet packet encapsulated in the WLAN packet. Accordingly, a variable number of packet descriptors is envisioned, as indicated by the ellipsis in Fig. 15. Packet descriptors 1530 and 1540 are shown, with each of the packet descriptors 1530 and 1540 including a packet flag (1550 and 1555, respectively) and a packet length (1560 and 1565, respectively).

The sequence number (1510) provides a sequence identifier for the encapsulated data, which allows the recipient to acknowledge receipt of the transmission. The ack number provides an acknowledgement for previously received data. The total packet number is the number of Ethernet packets that are encapsulated in the WLAN packet.

The packet flags (1550, 1555) indicate whether the associated Ethernet packet is a complete packet. Given that the time slot has a fixed duration, it is possible that an entire Ethernet packet might not fit into a given WLAN packet. Accordingly, in particular implementations it is expected that the first and last Ethernet packets will typically be incomplete in any given WLAN packet. The packet length (1560, 1565) indicates the length of that particular Ethernet packet.

Continuing with the process 1100, in the implementation of Fig. 10, the operation 1150 may be performed by, for example, the PADM 1016 of the modem 1010. Other implementations may perform the operation 1150 in, for example, the bridge, the Ethernet interface,
the WLAN interface, another intermediary component other than the PADM, a component above the bridge, or in a combination of components. As should be clear, the component(s) performing the operation 1150 may be implemented in, for example, software (such as a program of instructions), hardware (such as an IC), firmware (such as firmware embedded in a processing device), or a combination.

Additionally, the PADM may be located in a different position within the modem (such as, for example, above the bridge or between the Ethernet interface and the bridge), within one of the interfaces or the bridge, and/or be distributed among multiple components.

The process 1100 further includes the modem sending the encapsulated packet to the AP over a cable (1160). The sent packet is intended for receipt by the router. The cable may include, for example, a coaxial cable, a fiber optic cable, or other wired transmission medium.

In a particular implementation, when a modem's uplink timeslot arrives, the modem will gather packets from the ingress queue and put them into one big WLAN packet. The WLAN packet is not bigger than the biggest packet that the timeslot allows. Conversely, when the timeslot arrives, if the WLAN packet is not big enough to fill the duration of the fixed timeslot, then one implementation still sends the (smaller) WLAN packet, whereas another implementation sends NULL data.

Referring to Fig. 16, a process 1600 depicts a process for receiving encapsulated packets, decapsulating the packets, and delivering the constituent packets. This process 1600 is also referred to as an uplink reception process.

The process 1600 includes an AP receiving an encapsulated packet from a modem over a WLAN interface
In the implementation of Fig. 10, the AP 1030 receives the encapsulated packet from the modem 1010. The packet is received at the WLAN interface 1075 over the cable network 1040 (such as a coaxial cable network).

The AP decapsulates the received packet to extract the constituent packets that make up the encapsulated packet (1630). In the implementation of Fig. 10, the WLAN interface 1075 sends the received (encapsulated) packet to the PADM 1076. The PADM 1076 performs decapsulation and provides the constituent Ethernet packets to the bridge 1074. Decapsulation is performed by examining, for example, the total packet number 1520, and the packet flag (for example, packet flag 1550) and packet length (for example, packet length 1560) of each packet descriptor (for example, packet descriptor 1530).

By examining such data, the PADM 1076 is able to determine where each of the constituent packets starts and ends.

In particular, the PADM 1076 examines each constituent packet to ensure that the constituent packet is a complete Ethernet packet. If the constituent Ethernet packet is not complete, then the PADM 1076 retains the incomplete packet and waits until the rest of the Ethernet packet is received (presumably in a subsequent encapsulated packet). When the rest of the Ethernet packet is received, the PADM 1076 assembles the complete Ethernet packet and forwards the complete Ethernet packet to the bridge 1074.

Referring to Fig. 17, the above implementation of the operation 1630 is depicted in a diagram 1700 for a received encapsulated packet 1710. For simplicity, the received encapsulated packet 1710 is assumed to be the same as the transmitted packet described with reference to Fig. 14. However, it is understood that variations
between a transmitted packet and a received packet might occur in practice. The received packet 1710 includes the WLAN header 1440, the EIW header 1430, and the string of constituent Ethernet packets 1420.

As the PADM 1076 processes the received packet 1710, if a constituent Ethernet packet is complete, then the packet (for example, a packet 1720) is provided to the bridge 1074. If a constituent Ethernet packet is incomplete, then the incomplete packet is stored in a waiting queue 1730 (which need not be located in the PADM 1076) until the rest of the packet arrives. The diagram 1700 shows an incomplete packet 1740 being stored in the waiting queue 1730. This may occur, for example, if an Ethernet packet spans two WLAN packets. When the packet is complete, the packet is sent to the bridge 1074. Note that a WLAN packet may include, for example, one complete Ethernet packet and one partial Ethernet packet.

Referring to Fig. 18, to further describe the decapsulation process 1130, a PADM 1750 is depicted that provides an implementation of either of the PADMs 1016 or 1076. The PADM 1750 includes an encapsulator 1760 and a decapsulator 1770. The encapsulator 1760 and the decapsulator 1770 are communicatively coupled to a bridge and a WLAN interface. Given the components of the PADM 1750, the PADM 1750 may, more specifically, be referred to as a packet encapsulation/decapsulation module.

In operation, the encapsulator 1760 accepts Ethernet packets from the bridge and encapsulates the Ethernet packets, as described above. The encapsulated data is then provided to the WLAN interface.

In operation, the decapsulator 1770 receives encapsulated data from the WLAN interface. The decapsulator 1770 decapsulates the received data as
described above, and provides the decapsulated data to the bridge.

Clearly, other implementations are possible and envisioned. For example, another implementation combines an encapsulator and a decapsulator. Yet another implementation uses a virtual Ethernet interface feature of Linux.

Note that other implementations of an AP or a modem send an encapsulated packet from a WLAN interface directly to a bridge. The bridge determines that the packet is encapsulated and sends the packet to a PADM.

Continuing with the process 1600, the AP determines that the constituent packets are to be sent to a router (1640). This operation (1640) may be performed, as with many operations, at a different point in the process 1600. In the implementation of Fig. 10, the bridge 1074 determines that the packets are to be sent to the router 1090.

The AP then sends the constituent packets to the router over an Ethernet interface (1650). In the implementation of Fig. 10, the bridge 1074 sends the constituent packets to the Ethernet interface 1077, which sends the packets to the router 1090 over the Ethernet network 1082.

The router receives (1060) and processes (1070) the packets. Processing may include, for example, sending the packets, or a portion thereof, to a further destination such as a web site with which a host is communicating or attempting to communicate. Further, in implementations in which an encapsulated packet includes Ethernet packets from multiple hosts, the router may send the underlying information to multiple web sites.

Referring to Fig. 19, a process 1800 depicts a process for receiving packets at an AP from a router.
The packets are encapsulated, and the encapsulated packets is transmitted from the AP. The transmitted encapsulated packet is intended for receipt by a modem, and the constituent packets are intended for eventual delivery from the modem to one or more hosts. This process 1800 is also referred to as a downlink transmission process.

The process 1800 includes a router receiving one or more packets intended for one or more hosts (1820), and the router sending the received packet(s) to an AP (1830). The router may receive packets from, for example, one or more web sites that are attempting to communicate with one or more hosts. In the implementation of Fig. 10, the router 1090 receives packets from the Internet 1095. The router 1090 then sends the received packets over the Ethernet network 1082 to the Ethernet interface 1077 of the AP 1030.

The AP determines that at least one received packet is to be sent over a WLAN interface to the modem (1840). In the implementation of Fig. 10, the Ethernet interface 1077 routes the received packets (which are Ethernet packets) to the bridge 1074. The bridge 1074 determines that a packet is to be sent over the WLAN interface 1075 to, for example, the modem 1010.

The AP encapsulates multiple packets for transmission to the modem, including the one or more received packets (1850). Note that the multiple packets are all received from the router, but may have been received at the router from one or more different sources (for example, different web sites). Further, the encapsulation may include the packet(s) received in operation 1820 and packets received earlier and stored in a queue.
Regarding the operation 1850, in the implementation of Fig. 10, the bridge 1074 forwards the received packet(s) to the PADM 1076. The PADM 1076 queues the received packet(s), along with other packets intended for (for example) the modem 1010 and forms an encapsulated WLAN packet for the available downlink timeslot for the modem 1010. The PADM 1076 maintains a separate queue for each modem (also referred to as a station), including a first queue for the modem 1010 and a second queue for the modem 1020. The encapsulation is as described earlier in describing the PADM 1016 in conjunction with Figs. 11-15.

The AP sends the encapsulated packet to the modem over a cable connection, intended for eventual delivery to one or more hosts (1860). In the implementation of Fig. 10, the PADM 1076 prepares a WLAN packet for each of the modems 1010 and 1020 in a round-bin manner. The PADM 1076 then supplies the prepared WLAN packets to the WLAN interface 1075 for insertion into the corresponding downlink time-slots in the TDF superframe structure. The WLAN interface 1075 then transmits the WLAN encapsulated packets to the modems 1010 and 1020, using the TDF superframe structure.

Referring to Fig. 20, a process 1900 depicts a process for receiving encapsulated packets, decapsulating the packets, and delivering the constituent packets. This process 1900 is also referred to as a downlink reception process.

The process 1900 includes a modem receiving an encapsulated packet from an AP over a WLAN interface (1920). In the implementation of Fig. 10, the modem 1010 receives the encapsulated packet at the WLAN interface 1017 over a cable network 1040 (such as a coaxial cable network).
The modem then decapsulates the received packet to extract the constituent packets that make up the encapsulated packet (1930). In the implementation of Fig. 10, the PADM 1016 performs decapsulation of the WLAN packet and provides the constituent Ethernet packets to the bridge 1014. The decapsulation may be performed, for example, as described earlier for the PADM 1076 in the discussion of Figs. 16-18.

The modem determines that the constituent packets are to be sent to one or more intended host recipients (1940). This operation (1940) may be performed, as with many operations, at a different point in the process 1900. For example, the operation 1940 may be performed in conjunction with either of operations 1930 or 1950. In the implementation of Fig. 10, the bridge 1014 determines that the packets are to be sent to the host(s).

The modem then sends the constituent packets to the host(s) over an Ethernet interface (1950). In the implementation of Fig. 10, the bridge 1014 sends the constituent packets to the Ethernet interface 1015, which sends the packets to one or more of the host1 1054 and the host2 1056 over the Ethernet network 1052.

The one or more hosts receive (1960) and process (1970) the packets. Processing may include, for example, a personal computer storing a multi-media file received over the Internet, or a personal digital assistant (PDA) displaying an electronic message (also received over the Internet) for viewing and interaction by a user.

Features and aspects of described implementations may be applied to various applications. Applications include, for example, individuals using host devices in their homes to communicate with the Internet using an Ethernet-over-cable communication framework, as described above. However, the features and aspects herein
described may be adapted for other application areas and, accordingly, other applications are possible and envisioned. For example, users may be located outside of their homes, such as, for example, in public spaces or at their jobs. Additionally, protocols and communication media other than Ethernet and cable may be used. For example, data may be sent and received over (and using protocols associated with) fiber optic cables, universal serial bus (USB) cables, small computer system interface (SCSI) cables, telephone lines, digital subscriber line/loop (DSL) lines, satellite connections, line-of-sight connections, and cellular connections.

The implementations described herein may be implemented in, for example, a method or process, an apparatus, or a software program. Even if only discussed in the context of a single form of implementation (for example, discussed only as a method), the implementation of features discussed may also be implemented in other forms (for example, an apparatus or program). An apparatus may be implemented in, for example, appropriate hardware, software, and firmware. The methods may be implemented in, for example, an apparatus such as, for example, a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processing devices also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants ("PDAs"), and other devices that facilitate communication of information between end-users.

Implementations of the various processes and features described herein may be embodied in a variety of different equipment or applications, particularly, for example, equipment or applications associated with data.
transmission and reception. Examples of equipment include video coders, video decoders, video codecs, web servers, set-top boxes, laptops, personal computers, and other communication devices. As should be clear, the equipment may be mobile and even installed in a mobile vehicle.

Additionally, the methods may be implemented by instructions being performed by a processor, and such instructions may be stored on a processor-readable medium such as, for example, an integrated circuit, a software carrier or other storage device such as, for example, a hard disk, a compact diskette, a random access memory ("RAM"), or a read-only memory ("ROM"). The instructions may form an application program tangibly embodied on a processor-readable medium. As should be clear, a processor may include a processor-readable medium having, for example, instructions for carrying out a process.

As should be evident to one of skill in the art, implementations may also produce a signal formatted to carry information that may be, for example, stored or transmitted. The information may include, for example, instructions for performing a method, or data produced by one of the described implementations. Such a signal may be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of spectrum) or as a baseband signal. The formatting may include, for example, encoding a data stream, packetizing the encoded stream, and modulating a carrier with the packetized stream. The information that the signal carries may be, for example, analog or digital information. The signal may be transmitted over a variety of different wired or wireless links, as is known.

A number of implementations have been described. Nevertheless, it will be understood that various
modifications may be made. For example, elements of different implementations may be combined, supplemented, modified, or removed to produce other implementations. Additionally, one of ordinary skill will understand that other structures and processes may be substituted for those disclosed and the resulting implementations will perform at least substantially the same function(s), in at least substantially the same way(s), to achieve at least substantially the same result(s) as the implementations disclosed. Accordingly, these and other implementations are contemplated by this application and are within the scope of the following claims.
CLAIMS

1. A method comprising:
   receiving (1130, 1820) a packet from a first source, the packet from the first source having a particular format;
   receiving (1130, 1820) a packet from a second source, the packet from the second source having the particular format; and
   encapsulating (1150, 1850) the packet from the first source and the packet from the second source into an encapsulated packet having a different format.

2. The method of claim 1 wherein:
   the different format is adapted for wireless transmission, and
   the method further comprises transmitting the encapsulated packet over a wired communication medium.

3. The method of any of claims 1 or 2 wherein:
   the particular format comprises an Ethernet format, the different format comprises a WLAN format, and

4. The method of claim 2 wherein the wired communication medium comprises coaxial cable.

5. The method of any of claims 2 to 4 wherein transmitting the encapsulated packet comprises transmitting the encapsulated packet in a slot of a time-division multiplexing scheme.

6. The method of claim 5 wherein encapsulating comprises encapsulating the packet from the first source and the packet from the second source along with a
limited portion of a third packet, such that the encapsulated packet includes only a limited portion of the third packet, and the limited portion of the third packet allows the encapsulated packet to fill the slot of the time-division multiplexing scheme.

7. The method of any of claims 2 to 6 wherein transmitting the encapsulated packet comprises transmitting the encapsulated packet according to a frequency-division multiplexing scheme on a signal carrying additional data, the scheme providing that the encapsulated packet be transmitted in a frequency range for the encapsulated data and that the additional data be transmitted in a different frequency range.

8. The method of claim 7 wherein the additional data comprises television data.

9. The method of any of claims 1 to 8 wherein:
   the packet from the first source includes data intended for receipt by a first destination, and
   the packet from the second source includes data intended for receipt by a second destination.

10. The method of any of claims 1 to 9 wherein the encapsulating provides increased throughput for at least the first source.

11. The method of claim 1 wherein:
   the particular format comprises an Ethernet format, the different format comprises a WLAN format adapted for wireless transmission,
   the method further comprises transmitting the encapsulated packet over a coaxial cable,
transmitting the encapsulated packet comprises
transmitting the encapsulated packet

(a) in a slot of a time-division multiplexing scheme, and

(b) according to a frequency-division multiplexing scheme on a signal carrying television data, the frequency-division multiplexing scheme providing that the encapsulated packet be transmitted in a frequency range for the encapsulated data and that the television data be transmitted in a different frequency range.

12. The method of claim 7-11 wherein the frequency-division multiplexing scheme is compliant with a time divisional function protocol.

13. An apparatus comprising:
   a receiver (1015, 1077) to receive a packet from a first source and to receive a packet from a second source, the packet from the first source and the packet from the second source having a particular format; and
   a processor (1016, 1076) to encapsulate the packet from the first source and the packet from the second source into an encapsulated packet having a different format.

14. An apparatus comprising:
   means (1015, 1077) for receiving a packet from a first source and receiving a packet from a second source, the packet from the first source and the packet from the second source having a particular format; and
   means (1016, 1076) for encapsulating the packet from the first source and the packet from the second source into an encapsulated packet having a second format.
15. An apparatus comprising a processor-readable medium including instructions stored on the processor-readable medium for performing at least the following:

receiving (1130, 1820) a packet from a first source, the packet from the first source having a particular format;

receiving (1130, 1820) a packet from a second source, the packet from the second source having the particular format; and

encapsulating (1150, 1850) the packet from the first source and the packet from the second source into an encapsulated packet having a different format.

16. The apparatus of claim 15 wherein the instructions form an application program tangibly embodied on the processor-readable medium.

17. A signal structured to carry an encapsulated packet having a given format, the signal comprising:

a first portion structured to carry a part (1352) of the encapsulated packet that includes a packet from a first source, the packet from the first source having a particular format that is different from the given format; and

a second portion structured to carry a part (1353) of the encapsulated packet that includes a packet from a second source, the packet from the second source having the particular format.

18. The signal of claim 17 further comprising a portion (1560, 1565) structured to carry information indicating:

a length of the part of the encapsulated packet that includes the packet from the first source, and
a length of the part of the encapsulated packet that includes the packet from the second source.

19. The signal of claims 17-18 wherein the signal represents digital information.

20. The signal of claims 17-19 wherein the signal is an electromagnetic wave.

21. The signal of claims 17-20 wherein:
   
   the given format of the encapsulated packet comprises a WLAN format,

   the particular format of the packets from the first source and the second source comprises an Ethernet format,

   the first portion and the second portion are configured for transmission within a slot of a time-division multiplexing scheme,

   the first portion and the second portion occupy a particular frequency range in a frequency-division multiplexing scheme, and

   the signal further comprises a television portion that occupies a different frequency range in the frequency-division multiplexing scheme.

22. A method comprising:

   receiving (1620, 1920) an encapsulated packet having a given format, the encapsulated packet including a packet from a first source and including a packet from a second source, the packet from the first source and the packet from the second source having a format different from the given format;

   extracting (1630, 1930), from the encapsulated packet, the packet from the first source; and
extracting (1630, 1930), from the encapsulated packet, the packet from the second source.

23. The method of claim 22 wherein:

- the different format comprises an Ethernet format,
- the given format comprises a WLAN format adapted for wireless transmission, and
- receiving the encapsulated packet comprises receiving the encapsulated packet (1) over a coaxial cable, (2) in a slot of a time-division multiplexing scheme, (3) in a particular frequency range of a frequency-division multiplexing scheme, wherein television data occupies a different frequency range of the frequency-division multiplexing scheme.

24. An apparatus comprising:

- a receiver (1017, 1077) to receive an encapsulated packet having a given format, the encapsulated packet including a packet from a first source and including a packet from a second source, the packet from the first source and the packet from the second source having a format different from the given format; and
- a processor (1016, 1076) to extract, from the encapsulated packet, the packet from the first source and the packet from the second source.

25. An apparatus comprising:

- means (1017, 1077) for receiving an encapsulated packet having a given format, the encapsulated packet including a packet from a first source and including a packet from a second source, the packet from the first source and the packet from the second source having a format different from the given format; and
means (1016, 1076) for extracting, from the encapsulated packet, the packet from the first source and the packet from the second source.

26. An apparatus comprising a processor-readable medium including instructions stored on the processor-readable medium for performing at least the following:
   receiving (1620, 1920) an encapsulated packet having a given format, the encapsulated packet including a packet from a first source and including a packet from a second source, the packet from the first source and the packet from the second source having a format different from the given format;
   extracting (1630, 1930), from the encapsulated packet, the packet from the first source; and
   extracting (1630, 1930), from the encapsulated packet, the packet from the second source.
Fig. 4

Flowchart:

Start

- Find TDF Synchronization frames before a timeout?
  - Yes: Enter TDF mode
  - No:
    - Scan for 802.11 network. Find it?
      - Yes: Enter 802.11 mode
      - No: Continue scanning for 802.11 network.
Fig. 5
Fig. 6

Fig. 7

Fig. 8
FIG. 11

MODEM CONNECTS TO AP

HOST(S) SENDS PACKET(S) FOR RECEIPT BY ROUTER

MODEM RECEIVES PACKET(S) FROM HOST(S)

MODEM DETERMINES THAT PACKET(S) IS (ARE) TO BE SENT OVER WLAN INTERFACE

MODEM ENCAPSULATES PACKETS FOR ROUTER

MODEM SENDS ENCAPSULATED PACKET TO AP OVER CABLE, INTENDED FOR RECEIPT BY ROUTER

FIG. 12

Ethernet Header  Data

WLAN mac Header  Data  FCS
AP RECEIVES ENCAPSULATED PACKET OVER WLAN INTERFACE FROM MODEM

AP DECAPSULATES RECEIVED PACKET INTO CONSTITUENT PACKETS

AP DETERMINES THAT CONSTITUENT PACKETS ARE FOR ROUTER

AP SENDS PACKETS TO ROUTER OVER ETHERNET INTERFACE

ROUTER RECEIVES PACKETS

ROUTER PROCESSES RECEIVED PACKETS

FIG. 16
FIG. 18
ROUTER RECEIVES PACKET(S) INTENDED FOR A HOST

ROUTER SENDS PACKET(S) TO AP

AP DETERMINES THAT PACKET(S) IS(ARE) TO BE SENT OVER WLAN INTERFACE

AP ENCAPSULATES PACKETS FOR MODEM

AP SENDS ENCAPSULATED PACKET TO MODEM OVER CABLE, INTENDED FOR DELIVERY TO HOST(S)

FIG. 19
MODEM RECEIVES ENCAPSULATED PACKET OVER WLAN INTERFACE FROM AP

MODEM DECAPSULATES RECEIVED PACKET INTO CONSTITUENT PACKETS

MODEM DETERMINES INTENDED HOST RECIPIENT(S) OF PACKETS

MODEM SENDS PACKETS TO HOST(S) OVER ETHERNET INTERFACE

HOST(S) RECEIVES PACKETS

HOST(S) PROCESS RECEIVED PACKETS

FIG. 20
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

H04L 29/02 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04LH04B G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI, EPDOC, PAJ, CNPAT, CNKI; encapsulat+ format ETHERNET WLAN multiplex data packet message transmission

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>CN1691666A, (HUawei TECH Co LTD [CN]) 02 Nov. 2005 (02.11.2005) claims 1-10, description page 8, line 3-page 9, line 18, figures 1-2</td>
<td>1, 3, 5, 9, 10, 12-20, 22, 24-26</td>
</tr>
<tr>
<td>X</td>
<td>CN1642116A, (HUawei TECH Co LTD [CN]) 20 Jul. 2005 (20.07.2005) claims 1-16, figures 1-4</td>
<td>1, 3, 5, 9, 10, 12-20, 22, 24-26</td>
</tr>
<tr>
<td>A</td>
<td>US2006215653A1, (KOOT-I) KootSTRA L S; (LAVI-I) LAVIgNE B E; (WATT-I) WATTS JM; (WICK-I) WICKERaad J A) 28 Sep. 2006 (28.09.2006) the whole document</td>
<td>1-26</td>
</tr>
</tbody>
</table>

- The circle indicates that the documents are listed in the continuation of Box C. 
- "X" See patent family annex.

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention.
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone.
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family.

Date of the actual completion of the international search: 05 Apr. 2008 (05.04.2008)

Name and mailing address of the ISA/CN:
The State Intellectual Property Office, the P.R.China
6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China 100088
Facsimile No. 86-10-62019451

Form PCT/ISA/210 (second sheet) (April 2007)


Authorized officer: LIANG Nianshun
Telephone No. (86-10)624 11273
# INTERNATIONAL SEARCH REPORT

Information on patent family members

<table>
<thead>
<tr>
<th>Patent Documents referred in the Report</th>
<th>Publication Date</th>
<th>Patent Family</th>
<th>Publication Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>£51691666A</td>
<td>02.11.2005</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>CN1642116A</td>
<td>20.07.2005</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>WO2005006751A2</td>
<td>20.01.2005</td>
<td>FR2857197A1</td>
<td>07.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FR2857197B1</td>
<td>14.10.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP1639819A2</td>
<td>29.03.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US2006242668A1</td>
<td>26.10.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NO20060551A</td>
<td>02.02.2006</td>
</tr>
<tr>
<td>JS2006215653A1</td>
<td>28.09.2006</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td>JP2002281060A</td>
<td>27.09.2002</td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>

Form P.CT/ISA/210 (patent family annex) (April 2007)