

WIRELESS PACKET DATA TRANSMISSION SYSTEM WITH SIGNAL VALIDITY DETECTOR

The present application relates to digital transmission of data. In particular, the application relates to a decoding method and to a decoding device.

The Manchester protocol provides an encoding and decoding protocol for digital signals. This is often utilized in communication systems where a minimum transition density is required, or where a direct current (DC) balanced signal is required. In accordance to the Manchester protocol, the digital signals are encoded using an encoder and are decoded using a decoder.

The Manchester encoded signal comprises a self-clocking code with a minimum of one and a maximum of two-level transitions per bit. The code assigns a data of zero, which is logic low, to a low-to-high transition, and assigns a data of one, which is logic high, to a high-to-low transition. For data of two identical consequent bits, the code ignores an extra level transition.

US 7,133,482 B2 shows a hardware structure of a data packet decoder for the Manchester protocol. The use of central bit transitions and time measurements between transitions is used for improving the Manchester decoding. This method depends on the incoming Manchester encoded data for decoding. Any erroneous packets could be discarded at the higher layers.

US 6,628,212 B1 shows a Manchester decoder that uses oversampling techniques. These techniques can be time consuming as the techniques depend on the incoming Manchester encoded data

for decoding and erroneous packets could be discarded at the higher communication layers only.

It is an objective of this application to provide an improved communication system. The communication system can support digital signal. In particular, the application provides a data packet based communication for wireless systems that can use infrared signals. The wireless systems can be used in indoor applications.

One of the main thoughts of the application is that reducing or eliminating processing time of "lost" data packets during packet-based communication can improve data throughput and efficiency. It is believed that the packets are often considered "lost" due to non-detection of the packets.

To detect the "lost" packets, the application provides a signal validity controller for a signal decoder or receiver. The signal validity controller receives the same data signal as the signal decoder and it performs statistical operations on the data signal. The operations generate validity information about the signal in relation to data packets that are contained in the signal. The validity information can be used for processing the data packets at a very low communication layer, thereby reducing time wasted on processing unwanted data.

The application provides a receiver device of a data packet communication network. The receiver device comprises a receiver unit, a decoder unit, and a data validity module.

The receiver unit is used for receiving one or more data packets that are encoded at a transmitter side and that are

then transmitted over a wireless medium to the receiver unit. The data packets can be encoded with the Manchester coding scheme. The encoding often allows efficiency transmission of the data packets over the wireless medium.

5

The decoder unit is used for decoding the received data packets to its form.

Referring to the data validity module, it comprises a data validity unit and a data validity controller unit.

10

The data validity unit also receives the transmitted data packets and it is used for generating data validity information of the received data packets. Several techniques can be used to determine that the received data packets are valid.

15

The data validity controller unit is connected with the decoder unit and which causes the decoder unit to abort the decoding of the data packet if the data validity information indicates that the data packet is invalid. The data packet can be corrupted or damaged during its transmission over a transfer medium.

20

This aborting of data packet has obvious advantages of saving processing time. System resources are saved from processing invalid data packets. The decoder unit can be adapted for decoding Manchester data packet or other decoding scheme.

25

The application provides a data validity module for a receiver device that receives data packets. The data validity module includes a data validity unit and a data validity controller unit. The data validity unit is used for generating data validity information of one or more data packets. The

30

data validity controller unit, which comprises an output connection for a receiver decoder unit that decodes the data packet. The data validity controller unit outputs a predetermined value indicating the data validity information of the data packet.

The data validity controller unit includes an output connection for a receiver decoder unit that decodes the data packet, wherein the data validity controller unit outputs a predetermined value indicating the data validity information.

The application provides a method of operating a receiver device of a data packet communication network.

The method comprises a step of receiving least one or more encoded data packets. The received encoded data packets are then decoded according to a predetermined coding scheme. After this, validity information of the data packets is generated using one of several possible techniques. The decoding of one or more data packets is aborted if the validity information indicates that the data packets are invalid.

The aborting allows system resources to be saved from or not used for processing invalid data packets. One invalid data packet can also caused associated data packets to be aborted when the associated data packets cannot be decoded or processed due to the invalid data packet.

The received data packets can be encoded with a Manchester coding scheme or other coding scheme.

The method can include a step of receiving a next data packet directly after the aborting or stopping the decoding of data

packet. In this case, there is no or reduced lost of system time due to invalid data packets.

The aborted data packet is not often transmitted to next
5 layer. The next layer is then saved or is prevented from processing invalid data packets.

The decoding can comprise one or more possible steps. The possible steps include receiving a preamble and a start of
10 frame limiter (SFD) of the data packet or getting length value from a Physical Layer Convergence Procedure (PLCP) header of the data packet. The step can also include checking Cyclic Redundancy Check (CRC) value of the data packet or sending the packet data from a Physical layer to a Media Ac-
15 cess Control (MAC) layer.

The application provides a method of operating a data validity module of a data packet receiver device. The method comprises a step of generating validity information of a (en-
20 coded) data packet and outputting a predetermined value indicating the data validity information.

In short, the application has the advantage of improving efficiency of a communication system in terms of both through-
25 put and use of energy. The system prevents erroneous data packets from going into upper communication layers for further processing. This thereby saves or conserves system resources for processing next incoming data packets.

30 Further processing of the erroneous data packets can consume system resources such that the system is prevented from receiving the next incoming data packets. A retransmission of these non-processed or non-detected data packets may be trig-

gered. This would consume transmission bandwidth. Hence, the application increases usable bandwidth by reducing overhead in packet transmissions. This is especially of significance in cases of already congested systems.

5

In a generic sense, the application can be applied to any type of packet data transmission scheme beside the Manchester coding scheme.

10 In the following description, details are provided to describe the embodiments of the application. It shall be apparent to one skilled in the art, however, that the embodiments may be practised without such details.

15 Fig. 1 illustrates a local area network (LAN) that uses the Manchester encoder/decoder encoding,
Fig. 2 illustrates an improved LAN that includes the LAN of Fig. 1 and a signal validity device,
Fig. 3 illustrates steps of a receiver state machine at a
20 receiver side of the LAN of Fig. 1, and
Fig. 4 illustrates steps of an improved receiver state machine at a receiver side of the improved LAN of Fig. 2.

25 Figs. 1 to 4 have similar parts. The similar parts have same names or same part numbers. The description of the similar parts is hereby incorporated by reference, where appropriate, thereby reducing repetition of text without limiting the disclosure.

30

Fig. 1 shows a first local area network (LAN) 10 that uses the Manchester coding scheme. The LAN 10 has a transmitter

side 12 and a receiver side 13. A wireless medium 15 separate the transmitter side 12 from the receiver side 13.

The transmitter side 12 includes an upper layer wireless local area network (WLAN) module 17 that is communicatively connected to a Manchester encoder module 18 whilst the Manchester encoder module 18 is connected to a transmitter 19. The Manchester encoder module 18 comprises a Physical Layer Convergence Procedure (PLCP) preamble and Start of Frame Delimiter (SFD) device 22, PLCP header and Cyclic Redundancy Check (CRC) computation device 23 and a Manchester encoder device 24.

The receiver side 13 includes a receiver 26 that is connected to a Manchester decoder module 27. The Manchester decoder module 27 is communicatively connected to an upper layer WLAN module 28. The Manchester decoder module 27 comprises a Manchester decoder device 30 that communicatively connected to a send device 31.

20

Functionally, the transmitter side 12 is used for sending digital data signals via the wireless medium 15 to the receiver side 13.

25 Referring the transmitter side 12, the upper layer WLAN module 17 sends digital data packets to the Manchester encoder module 18 to encode for transmitting via the transmitter 19. The PLCP preamble and SFD device 22 generates a packet preamble segment and a packet start of frame delimiter segment for the data packets. The PLCP header and CRC computation device 30 23 generates a packet header segment and a packet CRC information segment for the digital data packets. The Manchester encoder device 24 encodes the preamble, the SFD segment, the

header segment, the CRC information segment, and the digital data using the Manchester coding scheme.

Referring the receiver side 13, the Manchester encoder module 5 27 receives the encoded data packets via the receiver 26. The Manchester decoder device 30 decodes the encoded packets in accordance to the Manchester coding scheme and it sends the decoded packets the send device 31. The send device 31 removes the header segment, checks the packets using the CRC 10 information segment, and sends the checked packets to the upper layer WLAN module 28.

The data packets are transmitted over the wireless medium 15 in single information bits that are formatted using the Manchester encoding scheme. The Manchester encoding scheme provides a modulation scheme that is suitable for low-cost radio-frequency transmission of digital data. The modulation scheme provides a simple method for encoding a digital serial of data of arbitrary bit pattern that is without any long 20 string of continuous zeros or ones. The encoded data also embeds an encoding clock rate.

These two characteristics allow the decoder device 30 to have low-cost data-recovery circuits for decoding the encoded data 25 that is transmitted with variable signal strengths and with imprecise, low-cost, data-rate clocks. The clock rate can be extracted from the encoded data by using a digital phase locked loop (DPLL).

30 The Manchester decoder device 30 uses a special signature for detecting of a start of a data packet. An end of the data packet does not carry a specific signature. Instead, no

change of signal is used for detecting the end of the data packet.

In a noisy environment, noise can exert a certain threshold
5 such that it is difficult to differentiate between noise and
valid data. The wireless medium 15 can have more noise than a
wired medium, as the wireless medium 15 can receive signals
from other networks or signal sources. In this respect,
transmission via a wireless standard, such as the WLAN 802.11
10 standard, is often less reliable as compared to the wired
transmission.

The noise may be present at the end of data packet in a man-
ner that prevents the decoder device 30 from detecting the
15 end of data packet. The noise signal thus causes the data
packet to be decoded incorrectly. This would cause the de-
coder device 30 to be left incorrectly in an active state in
which it still tries to decode the data packet. Some waiting
time is then required, for example x ns (nanoseconds), in or-
20 der for the decoder device 30 to come to the IDLE state that
is a normal state such that the receiver is able to receive
the next incoming data packet. During this waiting time, the
decoder device 30 is unable to decode any incoming data pack-
ets and these incoming data packets are then dropped or ig-
25 nored. This may trigger a retransmission of the dropped data
packet.

Considering the accumulative effects of noise signal and
these often-unnoticed effects, the system efficiency can de-
30 crease by a great margin.

Fig. 2 shows an improved LAN 35 that includes the LAN of Fig.
1 and a signal validity device 37.

The signal validity device 37 includes a signal validity detector 38 and a signal validity controller 39. The signal validity detector 38 is also called a data validity detector circuit or a data validity detector module. The signal validity detector 38 is incorporated in the receiver 26 whilst the signal validity controller 39 is incorporated in the Manchester encoder module of Fig. 1 to form an improved Manchester encoder module 40.

To support the digital transmission, the transmitter 19 incorporates an Analog Front End (AFE) and an Analog to Digital Converter (ADC) whilst the receiver 26 also incorporates an AFE and a Digital to Analog Converter (DAC).

Functionally, the Manchester decoder device 30 starts generating a clock signal as soon as it finds incoming data packet signals and it starts to pump or send 1's and 0's into a receiver state machine. The clock starts at a first edge of the incoming data packet signals.

The signal validity detector 38 acts to validate the incoming data packet signals and it sends validity control signals to the signal validity controller 39. The validity control signals are also called validity signals or data valid signals.

The validity can use statistical operations on the signal in order to generate information about the start and the end of each data packet, which is contained in the received signal. The statistical operations can include correlation analysis, such as auto-correlation functions. Other techniques, which are used for cognitive radio spectrum sensing, can be applied accordingly. Alternatively, the validation can use techniques

such as energy detection method, cyclostationary feature detection methods, or other advanced methods, to distinguish between noise and modulated signals.

5 The signal validity controller 39 controls the Manchester decoder device 30 and the send device 31 using the received validity control signals. The control can stop the Manchester decoder device 30 quickly if necessary, thus saving the precious processing time. This would in turn reduce occurrence
10 of retransmission due to "lost" or to non-detected data signals.

This is unlike other LANs where the Manchester decoder devices cannot be stopped. A waiting period is then required
15 for the entire LAN to come to the IDLE State. Once the data and the clock signals enters into a state machine or a Physical layer, it takes up precious clock cycles, which is in the order of magnitude of microseconds for it to come to the IDLE State. The Physical Layer, here, provides hardware for transmitting signals of a network.
20

A data packet format of WLAN system can designate the first 128 bits as preamble and the next 8 bits as Start of Frame Delimiter (SFD). This would take about 134 clock cycles for
25 the WLAN system to receive the SFD after it starts receiving the 1's and 0's from Manchester decoder 30. This translates into 134 clock cycles for the WLAN system to come back to the IDLE State before it can start receiving the next data packet signal. By utilizing the validity signal, the WLAN system can
30 come to the IDLE state before the WLAN system goes through 134 clock cycles. This would thus reduce stabilization time and errors that occur due to this. Indirectly, this translates into improved efficiency.

In short, by utilizing the validity signal that is provided by the signal validity detector 38, the LAN 35 can stopped further processing of the packet signal and the LAN 35 can be set to the IDLE State earlier for receiving further packets. This way of stopping the entire LAN would not be possible for other systems, which depend only on received encoded data.

Fig. 3 shows a flow chart 43 of steps or states of a receiver state machine or Physical layer state machine at a receiver side of the LAN of Fig. 1.

The receiver state machine processes received data signal through four check steps before a Media Access Control Protocol Data Unit (MPDU) is passed on to a Media Access Control (MAC) layer. These steps do not perform error-coding correction. In other words, an error of even a single bit can lead to a discarding of the data.

The flow chart 43 includes an Idle state 46, a Preamble & SFD state 47, a Length state 48, a CRC check state 49, a PLCP Protocol Data Unit (PPDU) length state 52, and an end of reception state 54.

Referring to the flow chart 43, the Preamble & SFD state 47 follows the Idle state 46. The Length state 48 follows the Preamble & SFD state 47. The CRC check state 49 follows the Length state 48. The PPDU length state 52 or the Idle state 46 follows the CRC check state 49. The end of reception state 54 follows the PPDU length state 52.

Preamble & SFD state 47

In this state 47, the state machine looks for a preamble, which has a "10101010" word pattern. Once it finds this pattern, it searches for a SFD pattern, which has a "10101011" word pattern. This state 47 takes about 128 clock cycles of processing time before it decides whether it needs to discard the data or not.

Length state 48

The state 48 stores up the next 16 bits after the SFD, which has a length value of the PPDU as the information. This state 48 takes 16 clock cycles of processing to determine the length value of the received data.

CRC check state 49

In this state 49, the length is used to generate a CRC data using a CRC generator in the receiver circuit. This generated CRC data is then compared with a CRC data that is within the incoming data packet to make sure that the received data packet is correct. This state 49 takes 16 clock cycles of processing time to get the length from the received data.

PPDU length state 52

This state 52 mainly pumps or transmits the data from the Physical layer into the MAC layer. The clock cycles taken for this depend on the length of the sent data. However, the PPDU data goes into the MAC and a predetermined error correction of the PPDU data is done. This means, that even few bits are missing, the missing bits can be recovered.

The MAC layer provides addressing and channel access control mechanisms for several terminals or network nodes to communicate within a multipoint network. Put simply, the MAC layer

also acts as an interface between a Logical Link Control (LLC) layer and physical layer of the network.

The state machines would go through all the states before coming to the Idle state 46 to process any received new incoming data, even if the received data is erroneous in nature.

Fig. 4 shows a flow chart 60 of steps or states of an improved receiver state machine at the receiver side of the improved LAN of Fig. 2.

The improved receiver state machine uses the validity signal that is described above to reduce largely redundancy. Analyzing each block or state of the state machine in detail would shows that a number of clock cycles can be gained by efficiently utilizing the data validity signal.

The flow chart 60 includes the Idle state 46 of Figure 3, an improved Preamble & SFD state 64, an improved Length state 65, an improved CRC check state 67, an improved PPDU length state 68, and the end of reception state 54 of Figure 3.

Referring to Fig. 4, the Idle state 46 can follow the improved Preamble & SFD state 64, the improved Length state 65, the CRC check state 67, the improved PPDU length state 68, and the end of reception state 54. The improved Preamble & SFD state 64 follows the Idle state 46. The improved Length state 65 follows the improved Preamble & SFD state 64. The CRC check state 67 follows the improved Length state 65. The improved PPDU length state 68 follows the CRC check state 67. The end of reception state 54 follows the improved PPDU length state 68.

To illustrate the flow chart 60, several cases are described below. The cases assume a packet structure that has a Preamble segment of 120 bits and a SFD segment of 8 bits followed
5 a Data Length segment of 16 bits, a CRC value segment of 16 bits and a MAC Protocol Data Unit (MPDU).

Case 1:

This case illustrates a working of the improved Preamble &
10 SFD state 64.

When the decoder 30 receives a preamble, the decoder 30 would start to recover the clock signal together with the data packet signal. In this case, say a few bits of the preamble
15 of 10 bits arrives at the receiver 26 and after this, nothing else arrives. Watching and using the validity signal, the state machine can quickly stop this process and not wait for the entire process to complete. The state machine would then move to the IDLE state, where the state machine can start
20 looking for the next data packet signal. For an extreme case, this manner can save 128 clock cycles. This is unlike other state machine that would wait for the completion of 128 clock cycles to detect the SFD pattern. If it then does not detect the SFD pattern, it would discard the data packet signal and
25 later move to the IDLE state.

Case 2:

This case illustrates a working of the improved Length state
65.

30

The improved Length state 65 calculates a CRC value of the length value of the received data packet. For any invalid or erroneous data that is coming into the state machine, the

calculated CRC value of the length of received data packet would not be equal to the received CRC value that is within the received data packet. By utilizing the data validity signal, the state machine can predict in advance whether the length value is wrong or not, thus saving the precious clock cycles. The number of clock cycles saved would depend on which bit of the length value the error occurred. This is unlike other state machine where invalid data is discarded only after the 16 clock cycles.

10

Case 3:

This case illustrates a working of the CRC check state 67.

The CRC value has 16 bits length. This state utilizes the data validity signal to make sure that the data is valid or not and thus saving few clock cycles. Again, the number of clock cycles saved would depend on which bit of the CRC value the error is detected or which data bit is invalid. This is different from other state machine where the state machine waits for 16 clock cycles to make sure that CRC is correct or not.

15
20

Case 4:

This case illustrates a working of the improved PPDU length state 68.

25

In this case, a data packet of 30 bytes length is sent to the receiver end. Due to unforeseen or unknown reasons only 20 bytes of the data arrives at the receiver side. Unlike other state machine that waits for the other 10 bytes to come - the probability of other bytes coming to the receiver is very rare - the improved PPDU length state 68, by utilizing the data validity signal, determines that the data arrived is in-

30

valid. By this way, several clock cycles are saved, which contribute to efficiency improvement of the entire system.

5 In summary, by utilizing the data signal and the data validity signal, the receiving state machine can efficiently stop unnecessarily different steps or states and it can spend more time in the IDLE state and be ready for receiving new valid data packets. This also prevents retransmissions and collisions from happening and thus improving overall efficiency of
10 the system.

Put simply, the embodiment uses the history of the signal for generating information on the validity of each received data packet. This is unlike other methods that only uses past information at a much higher communication layer that involves
15 much more computing power. The embodiment circumvents the lost of data packets. In a generic sense, the embodiment can be applied to other types of packet data transmission.

20 Although the above description contains much specificity, these should not be construed as limiting the scope of the embodiments but merely providing illustration of the foreseeable embodiments. Especially the above stated advantages of the embodiments should not be construed as limiting the scope
25 of the embodiments but merely to explain possible achievements if the described embodiments are put into practise. Thus, the scope of the embodiments should be determined by the claims and their equivalents, rather than by the examples given.

Reference

	10	local area network
	12	transmitter side
5	13	receiver side
	15	wireless medium
	17	upper layer WLAN module
	18	Manchester encoder module
	19	transmitter
10	22	PLCP preamble and SFD device
	23	PLCP header and CRC computation device
	24	Manchester encoder device
	26	receiver
	27	Manchester decoder module
15	28	upper layer WLAN module
	30	Manchester decoder device
	31	send device
	35	LAN
	37	signal validity device
20	38	signal validity detector
	39	signal validity controller
	40	Manchester encoder module
	43	flow chart
	46	Idle state
25	47	Preamble & SFD state
	48	Length state
	49	CRC check state
	52	PPDU length state
	54	an end of reception state
30	60	flow chart
	64	improved Preamble & SFD state
	65	improved Length state
	67	improved CRC check state

68 improved PPDU length state

Claims

1. A receiver device (40) of a data packet communication network (13), comprising
 - 5 a receiver unit (26) for receiving at least one data packets,
 - a decoder unit (30) for decoding the at least one received data packets and
 - a data validity module (37) comprising
 - 10 - a data validity unit (38) for generating data validity information of the at least one received data packet and
 - a data validity controller unit (39) which is connected with the decoder unit (30) and which
 - 15 causes the decoder unit (30) to abort the decoding of the at least one received data packet if the data validity information indicates that the data packet is invalid.
- 20 2. A receiver device (40) of claim 1 wherein the decoder unit (30) is adapted for decoding Manchester data packet.
3. A data validity module (37) for a data packet receiver device, the data validity module comprising
 - 25 a data validity unit (38) for generating data validity information of at least one data packet and
 - a data validity controller unit (39) which comprises an output connection for a receiver decoder unit
 - 30 (30) that decodes the data packet, wherein the data validity controller unit (39) outputs a predetermined value indicating the data validity information.

4. A method of operating a receiver device (40) of a data packet communication network (13), the method comprising receiving at least one data packet, decoding the at least one data packet, generating validity information of the at least one data packet, and causes the decoding of the at least one data packet to be abort if the validity information indicates that the at least one data packet is invalid.

5. A method of claim 4, wherein the at least one received data packet is encoded with a Manchester coding scheme.

6. A method of claim 4 or 5 further comprising receiving a next data packet.

7. A method of one of claims 4 to 6, wherein the aborted data packet is not transmitted to next layer.

8. A method of one of claims 4 to 7, wherein the decoding comprises receiving a preamble and a start of frame limiter (SFD) of the data packet.

9. A method of one of claims 4 to 7, wherein the decoding comprises getting length value from a Physical Layer Convergence Procedure (PLCP) header of the data packet.

10. A method of one of claims 4 to 9, wherein the decoding comprises checking Cyclic Redundancy Check (CRC) value of the data packet.

11. A method of one of claims 4 to 10, wherein
the decoding comprises sending the packet data from a
Physical layer to a Media Access Control (MAC) layer.

5

12. A method of operating a data validity module (37) of a
data packet receiver device (40), the method comprising
generating validity information of a (encoded) data
packet and

10

- outputting a predetermined value indicating the
data validity information.