SYSTEM AND METHOD FOR SYNCHRONISED TELEMATIC MUSIC COLLABORATION

TECHNICAL FIELD

[0001] This invention relates broadly to the field of audio or music telematics, and more specifically to a system and associated method for synchronised telematic music collaboration via a communications network, such as the internet.

BACKGROUND ART

[0002] The following discussion of the background art is intended to facilitate an understanding of the present invention only. The discussion is not an acknowledgement or admission that any of the material referred to is or was part of the common general knowledge as at the priority date of the application.

[0003] In the field of audio telematics, being the science of sending, receiving and storing audio and video information via telecommunication devices, live music performance has a low tolerance for network latency, with any latency of around 30ms deemed to be an upper limit of what is acceptable. Conventional audio telematics via wide-area networks (WAN), such as the internet, is typically limited to collaborators being within 1000kms of each other to maintain such latency to within acceptable limits.

[0004] During the recent global COVID-19 pandemic, where social distancing and isolation were required, there were numerous attempts by musicians to collaborate by connecting

over the internet to play their instruments together at the same time, but due to the latency of exchanging audio (and/or visual) streams, which are notoriously bandwidth expensive, such attempts were generally not very successful due to latency problems.

[0005] As is known in the art of audio telematics, the Musical Instrument Digital Interface (MIDI) is a technical standard that describes a communication protocol, digital interface, and electrical connectors that connect a wide variety of electronic musical instruments, computers, and related audio devices for playing, editing, and recording music. An associated file format for storing and exchanging such music data is also defined as part of the MIDI protocol. In general, a MIDI file is a set of instructions, e.g. for pitch or tempo, rather than an audio recording, and thus requires orders of magnitude less bandwidth compared to an equivalent recorded audio.

[0006] MIDI was developed so that electronic or digital musical instruments could communicate with each other and so that one instrument could control another. A single MIDI port can carry up to sixteen channels of MIDI data, each of which can be routed to a separate device. Each interaction with a key, button, knob or slider is converted into a MIDI event, which specifies musical instructions, such as a note's pitch, timing and loudness.

[0007] Similarly, the Standard MIDI File (SMF) is a file format that provides a standardised way for music sequences to be saved, transported, and opened in other processing systems. These files are intended for universal use and include such information as note values, timing and track names, with lyrics included as metadata. SMFs organise MIDI messages into

one or more parallel tracks and time-stamp the events so that they can be played back in sequence. A header contains the arrangement's track count, tempo and an indicator of SMF format the file uses. Accordingly, there is no error detection capability in MIDI protocol.

[0008] While one solution to MIDI timing problems when electronically transferring MIDI files is to mark MIDI events with the times they are to be played, and subsequently store them in a buffer in the MIDI interface ahead of time, this cannot be done with live, online music collaboration, as any buffering is counter-productive to latency reduction. Another conventional solution has been the development of RTP-MIDI, which is a protocol to transport MIDI messages within Real-time Transport Protocol (RTP) packets over computer-based IP networks, which supports sample-accurate synchronization for each MIDI message. However, a common issue with RTP-MIDI is again latency during transmission, with any latency compensation mechanisms offered thereby mainly applicable only to pre-recorded MIDI streams, rather than live online music collaboration.

[0009] Conventional approaches to addressing the problem of real time collaborative music performance include systems and method as described in **US 2007/0039449** to eJamming Inc., wherein method and apparatus are disclosed to permit real time, distributed performance by multiple musicians at remote locations, and for recording that collaboration. This disclosure relies on a recovery process, where sequence numbers are used to determine which MIDI messages have gone missing, and using a journal to recover the missing data.

[0010] A further conventional approach for transmitting/receiving musical sound control data via a

communication network, such as the Internet, is disclosed in US 2005/0172790 to Yamaha Corporation. In this disclosure, a transmitter distinguishes and classifies MIDI messages as system exclusive message that should not be lost, as opposed to 'other' MIDI messages that are less important. The system exclusive messages are then transmitted via TCP with high reliability, and the 'other' messages via UDP having low reliability.

[0011] Such conventional systems rely on timestamp data sent between collaborators to calculate and estimate latency for synchronisation purposes. However, such conventional approaches cannot account for data packet delays if there is not a timestamp, similar to the known shortcomings of RTP-MIDI.

[0012] Accordingly, Applicant has identified a need in the art for a full-duplex MIDI routing system operable via a wide-area network, such as the internet, that is reliable (over an unreliable network such as the internet) but has less complexity than RTP-MIDI, and which enables synchronised telematic music performance between at least two collaborators. The current invention was conceived with this goal in mind.

SUMMARY OF THE INVENTION

[0013] The skilled addressee is to appreciate that reference herein to 'the internet' generally refers to any suitable wide-area network (WAN) communications network and typically includes reference to a global system of interconnected computer networks that use the Internet Protocol suite (IP) to link processing devices worldwide. Such a network includes a network of networks that may consist of

private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies, a broad example of which is provided herein.

[0014] It is also to be appreciated that reference to UDP comprises reference to the User Datagram Protocol (UDP), being one of the core communication protocols of the Internet Protocol suite, and uses a simple connectionless communication model with a minimum of protocol mechanisms, without error checking and correction capabilities, that prioritises transmission time over data reliability. Conversely, reference herein to TCP comprises reference to the Transmission Control Protocol (TCP) as another one of the main protocols of the Internet protocol suite, and which is connection-oriented, where a connection between sender and receiver is established before data can be sent, adding transmission reliability and error checking but lengthens latency. In general, UDP is suitable for live and real-time data transmission, which TCP cannot support.

[0015] It is also to be appreciated that Network Time Protocol (NTP) is a networking protocol that allows the synchronization of system clocks between processing systems, as is known in the art of computer science. NTP is described in the specification *RFC 1305–Network Time Protocol (Version 3) Specification, Implementation and Analysis* by the Internet Activities Board of the Defense Advanced Research Projects Administration (DARPA).

[0016] It is also to be understood that reference herein to a ‘GUI’ refer to a Graphical User Interface, being a user interface that allows a user to interact with an electronic device, such as a terminal, processing or computing system

through manipulation of graphical icons, visual indicators, text-based typed command labels and/or text navigation, including primary and/or secondary notations, as is known in the art of computer science.

[0017] It is further to be appreciated that reference herein to 'real-time' is to be understood as meaning an instance of time that may include a delay typically resulting from processing, calculation and/or transmission times inherent in computer processing systems. These transmission and calculation times, albeit of generally small duration, do introduce some measurable delay, i.e. typically less than a second or within milliseconds, but such a delay is generally imperceptible by a user.

[0018] According to a first aspect of the invention there is provided a system for synchronised telematic music collaboration comprising:

at least two MIDI-enabled user processing systems arranged in signal communication with the internet; and

a control server arranged in signal communication with the internet, said remote server configured to:

i. upon request, create a shared collaboration session on each user processing system, each session linked to the control server by assigned UDP and TCP ports allocated to the collaboration session;

ii. establish a UDP MIDI data stream with each session via said allocated UDP port;

iii. establish a multiplexed TCP control MIDI data stream with each session via said allocated TCP port;

iv. configure each session periodically to transmit a local timestamp signal based on the Network Time Protocol (NTP); and

v. exchange all MIDI data for the session via both allocated UDP and TCP ports;

wherein real-time music is shareable between sessions via both UDP and TCP MIDI data streams with the TCP control MIDI data stream useable to synchronise said shared UDP and TCP MIDI data streams via UDP and TCP MIDI data stream and periodic local timestamp signal comparison.

[0019] In an embodiment, the control server, via the TCP control MIDI data stream, synchronises the shared MIDI data streams by determining latency between each session via the periodic local timestamp signals from each session and calibrating exchange of MIDI data between sessions accordingly.

[0020] In an embodiment, the control server, via the TCP control MIDI data stream, maintains an integrity and synchronisation of the UDP and TCP MIDI data streams by comparing the TCP MIDI data stream to the UDP MIDI data stream to identify missing or delayed MIDI UDP data.

[0021] In an embodiment, the at least two MIDI-enabled user processing systems and the control server are arranged in signal communication, via the internet, by means of a star network topology with said control server as a central hub.

[0022] In an embodiment, the control server creates the collaboration session by allocating UDP and TCP port numbers to each session to coordinate MIDI sharing between sessions.

[0023] In an embodiment, the control server routes 16 channels of MIDI per UDP and TCP port allocation, including 128 notes and 128 Control Change controllers per channel and additional MIDI signals as per the RFC 6295 RTP Payload Format for MIDI specification.

[0024] In an embodiment, the control server maintains the integrity of the UDP MIDI data stream, via the TCP control MIDI data stream, by prioritising MIDI 'note off', 'pitch bend' and 'Controller Change' messages between sessions.

[0025] In an embodiment, the control server is configured to generate an optional audio and/or visual feed for an online audience.

[0026] In an embodiment, the MIDI-enabled user processing system comprises a MIDI musical instrument.

[0027] In an embodiment, each MIDI-enabled user processing system comprises a software application configured to provide a GUI for the collaboration session established by the control server.

[0028] In an embodiment, the MIDI-enabled user processing system is preconfigured, via a software application, with local audio files and/or synthesisers that are trigged by the UDP MIDI data stream and/or TCP control MIDI data stream.

[0029] In an embodiment, the control server is configured to encrypt the TCP control MIDI data stream.

[0030] In an embodiment, the control server is configured to prepare and transmit a TDP MIDI audience data stream comprising a combination of all MIDI data streams from all

sessions that have been time-corrected for latency and delay between respective sessions.

[0031] According to a second aspect of the invention there is provided a method for synchronised telematic music collaboration, said method comprising the steps of:

by means of a control server, creating a collaboration session on at least two MIDI-enabled user processing systems via the internet, each user processing system linked to the control server by an assigned UDP and TCP port which is allocated MIDI channels, each user processing system configured periodically to transmit a local timestamp signal based on Network Time Protocol (NTP);

via the control server, establishing a UDP MIDI data stream with each session via the UDP port;

via the control server, establishing a multiplexed TCP control MIDI data stream with each session via the TCP port;

via the control server, exchange all MIDI data for the session via both allocated UDP and TCP ports; and

via the control server, synchronising all sessions by comparing both UDP and TCP MIDI data streams using said local timestamp signals;

wherein real-time music is shareable between sessions via both UDP and TCP MIDI data streams with the TCP control MIDI data stream useable to synchronise said shared UDP and TCP MIDI data streams via UDP and TCP MIDI data stream and periodic local timestamp signal comparison.

[0032] In an embodiment, the step of synchronising the UDP and TCP MIDI data streams comprises the control server determining latency between each session via the periodic local timestamp signals from each session and calibrating exchange of MIDI data between sessions accordingly.

[0033] In an embodiment, the step of synchronising the UDP and TCP MIDI data streams comprises the control server comparing the TCP MIDI data stream to the UDP MIDI data stream to identify missing or delayed MIDI UDP data.

[0034] In an embodiment, the step of creating a collaboration session between the at least two MIDI-enabled user processing systems and the control server comprises arranging the MIDI-enabled user processing systems and the control server in signal communication, via the internet, by means of a star network topology with said control server as a central hub.

[0035] In an embodiment, the step of creating the collaboration session via the control server comprises said control server allocating MIDI port numbers to each session to coordinate MIDI sharing between sessions.

[0036] In an embodiment, the step of creating the collaboration session via the control server comprises the control server routing 16 channels of MIDI per UDP and TCP port allocation, including 128 notes and 128 Control Change controllers per channel and additional MIDI signals as per the RFC 6295 RTP Payload Format for MIDI specification.

[0037] In an embodiment, the control server maintains the integrity of the UDP MIDI data stream, via the TCP control MIDI data stream, by prioritising MIDI ‘note off’, ‘pitch bend’ and ‘Controller Change’ messages between sessions.

[0038] In an embodiment, the method includes a step of generating an audio and/or visual feed for an online audience via the control server.

[0039] In an embodiment, the step of creating a collaboration session on each user MIDI-enabled processing system comprises installing a software application on such processing system, said software application configured to provide a GUI for the collaboration session.

[0040] In an embodiment, the MIDI-enabled user processing system is preconfigured, via the software application, with local audio files and/or synthesisers that are trigged by the UDP MIDI data stream and/or TCP control MIDI data stream.

[0041] In an embodiment, the method includes a step of encrypting the TCP control MIDI data stream via the control server.

[0042] In an embodiment, the method comprises the step of, via the control server, preparing and transmitting a TDP MIDI audience data stream comprising a combination of all MIDI data streams from all sessions that have been time-corrected for latency and delay between respective sessions.

[0043] According to a further aspect of the invention there is provided a computer programme product which, when executed by a suitable processing system, facilitates the performance of the method according to the second aspect of the invention above.

[0044] According to a further aspect of the invention there is provided a system and associated method for synchronised telematic music collaboration, substantially as herein described and/or illustrated.

BRIEF DESCRIPTION OF THE DRAWINGS

The description will be made with reference to the accompanying drawings in which:

Figure 1 illustrates a functional block diagram of an example processing system that can be utilised to embody or give effect to a particular embodiment of a MIDI-enabled user processing system or the control server, in accordance with aspects of the present invention;

Figure 2 illustrates an example network infrastructure that can be utilised to embody or give effect to a particular embodiment of a wide-area communications network, such as the internet, whereby MIDI-enabled user processing systems and the control server can be arranged in signal communication; and

Figure 3 is a diagrammatic overview representation of a broad example of a system for synchronised telematic music collaboration, in accordance with aspects of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

[0045] Further features of the present invention are more fully described in the following description of several non-limiting embodiments thereof. This description is included solely for the purposes of exemplifying the present invention to the skilled addressee. It should not be understood as a restriction on the broad summary, disclosure or description of the invention as set out above.

[0046] In the figures, incorporated to illustrate features of the example embodiment or embodiments, like reference

numerals are used to identify like parts throughout. Additionally, features, mechanisms and aspects well-known and understood in the art will not be described in detail, as such features, mechanisms and aspects will be within the understanding of the skilled addressee.

[0047] With brief reference firstly to Figure 3 of the accompanying drawings, there is shown a broad example of a system 10 for synchronised telematic music collaboration. As described in more detail below, such a system 10 comprises a number of constituent components, including at least two user processing systems 12.1 and a control server 18 that are operatively networked together by means of a suitable wide-area communications network, such as the internet 16, or the like, in order to fulfil their functions as part of the present invention.

[0048] In particular, the system 10 seeks to provide a duplex MIDI routing system operable over a WAN, such as the internet, and enabling synchronised telematic music performance between remote collaborators 12.2. In general, the system 10 is realised by means of a control server 18 configured with routing software, typically based on RTP MIDI, which is executed as part of a web-based remote server, such as Amazon™ Web services (AWS), or the like, i.e. runs in third-party virtual machines or cloud-based servers, etc. The system 10 further includes MIDI-enabled user processing systems 12.1 configured with session software with an associated user interface.

[0049] With reference firstly to Figures 1 and 2 of the accompanying drawings, by way of enabling background, there is shown a broad example of a processing system 100 that can be used, in different configurations as will be readily

apparent to the skilled addressee, to implement a suitable user processing system 12.1 and the control server 18. Similarly, Figure 2 shows a broad example of a networked communications system 200 whereby the respective processing systems 12.1 and 18 can be arranged in signal communication.

[0050] It is to be appreciated that any reference herein to “means” specifically includes any one or more of a computer programme product for use in a local or dispersed computing system, a computer readable modulated carrier signal for interpretation by a local or dispersed computing system, or a computer readable medium of instructions for enabling a local or dispersed computing system to provide such “means” within the context of the description. In addition, such “means” may further expressly comprise any of the hardware and/or software components, independently or in combination, provided for in the description below, as will be understood by the skilled addressee.

[0051] As is known in the art of computer programming, an application programming interface (API) is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. Means for facilitating any of the network communications or interactions between user processing systems 12.1 and the control server 18 may be facilitated via suitable APIs within processing system 100 or network 200, as will be readily apparent to the skilled addressee.

[0052] Broadly, in a general networked information or data communications system, a user has access to one or more terminals which are capable of requesting and/or receiving information or data from local or remote information sources.

In such a communications system, a terminal may be a type of processing system, computer or computerised device, a MIDI instrument, personal computer (PC), mobile, cellular or satellite telephone, mobile data terminal, portable computer, Personal Digital Assistant (PDA), pager, thin client, or any other similar type of digital electronic device. The capability or means of such a terminal to request and/or receive information or data can be provided by software, hardware and/or firmware. A terminal may include or be associated with other devices, for example a local data storage device such as a hard disk drive or solid-state drive, a musical instrument with MIDI digitiser, or the like.

[0053] An information source can include a server, or any type of terminal, that may be associated with one or more storage devices that are able to store information or data, for example in one or more databases residing on a storage device. The exchange of information (i.e. the request and/or receipt of information or data) between a terminal and an information source, or other terminal(s), is facilitated by a communication means. The communication means can be realised by physical cables, for example a metallic cable such as a telephone line, semi-conducting cables, electromagnetic signals, for example radio-frequency signals or infra-red signals, optical fibre cables, satellite links or any other such medium or combination thereof connected to a network infrastructure.

[0054] The Internet, which often serves as an enabling part of communications network 200, is the large-scale interconnection of public and private networks. The network infrastructure can include devices such as a telephone switch, base station, bridge, router, or any other such specialised network component, which facilitates the connection between a

terminal and an information source. Collectively, an interconnected group of terminals, communication means, infrastructure and information sources are referred to as a network. The network itself may take a variety of forms. For example, it may be a computer network, telecommunications network, data communications network, Local Area Network (LAN), Wide Area Network (WAN), wireless network, Internetwork, Intranetwork, the Internet and developments thereof, transient or temporary networks, combinations of the above or any other type of network providing for communication between computerised, electronic or digital devices. More than one distinct network can be provided, for example a private and a public network. A network as referenced in this specification should be taken to include any type of terminal or other similar type of electronic device, or part thereof, which is rendered such that it is capable of communicating with at least one other terminal.

[0055] The Internet protocol suite, commonly known as TCP/IP, is a framework for organizing the set of communication protocols used in the Internet and similar computer networks according to functional criteria. The foundational protocols in the suite are the Transmission Control Protocol (TCP), the User Datagram Protocol (UDP), and the Internet Protocol (IP). The Internet protocol suite provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed, and received. The skilled addressee will appreciate that other communication protocols may be used and are within the scope of the present invention.

[0056] In light of this background, a general processing system 100 of Figure 1 generally includes at least one processor 102, or processing unit or plurality of processors, memory 104, at least one input device 106 and at least one

output device 108, coupled together via a bus or group of buses 110. Typically, the processor 102 comprises any suitable processor or microcontroller configured to receive input, perform logical and arithmetical operations on a suitable instruction set, and provide output, as well as transitory and/or non-transitory electronic storage, such as memory 104 and storage device 114, or the like.

[0057] In certain embodiments, input device 106 and output device 108 could be the same device, e.g. a touchscreen. An interface 112 can also be provided for coupling the processing system 100 to one or more peripheral devices, for example interface 112 could be a PCI card or PC card, MIDI digitiser or synthesiser, or the like. At least one storage device 114 which houses at least one database 116 can also be provided. The memory 104 can be any form of memory device, for example, volatile or non-volatile memory, solid state storage devices, magnetic devices, etc. The processor 102 could include more than one distinct processing device, for example to handle different functions within the processing system 100.

[0058] Input device 106 receives input data 118 and can include, for example, a music keyboard, a pointer device such as a pen-like device or a mouse, audio receiving device for voice-controlled activation such as a microphone, data receiver or antenna such as a modem or wireless data adaptor, data acquisition card, a touchscreen for receiving tactile input, etc. Input data 118 could come from different sources, for example keyboard instructions in conjunction with data received via a network, as is known in the art, or the like. Output device 108 produces or generates output data 120 and can include, for example, a display device or monitor in which case output data 120 is visual, a printer in which case output data 120 is printed, a port for example a Universal Serial Bus

(USB) port, a peripheral component adaptor, a data transmitter or antenna such as a modem or wireless network adaptor, etc. Output data 120 could be distinct and derived from different output devices, for example a visual display on a monitor in conjunction with data transmitted to a network.

[0059] In use, the processing system 100 may be adapted to allow data or information to be stored in and/or retrieved from, via wired or wireless communication means, the at least one database 116. The interface 112 may allow wired and/or wireless communication between the processing unit 102 and peripheral components that may serve a specialised purpose, such as a MIDI instrument, or the like. The processor 102 receives instructions as input data 118 via input device 106 and can display processed results or other output to a user by utilising output device 108. More than one input device 106 and/or output device 108 can be provided. It should be appreciated that the processing system 100 may be any form of terminal, server, specialised hardware, or the like.

[0060] As described, the processing system 100 is generally part of a networked communications system 200, as shown in Figure 2. Processing system 100 could connect to network 202, for example the Internet or a WAN. Input data 118 and output data 120 could be communicated to other devices via network 202. Other terminals, for example, thin client 204, further processing systems 206 and 208, notebook computer 210, mainframe computer 212, PDA 214, pen-based computer 216, server 218, etc., can be connected to network 202. A large variety of other types of terminals or configurations could be utilised. The transfer of information and/or data over network 202 can be achieved using wired communications means 220 or wireless communications means 222. Server 218 can

facilitate the transfer of data between network 202 and one or more databases 224.

[0061] Other networks may communicate with network 202. For example, telecommunications network 230 could facilitate the transfer of data between network 202 and mobile or cellular telephone 232 or a PDA-type device 234, by utilising wireless communication means 236 and receiving/transmitting station 238. Satellite communications network 240 could communicate with satellite signal receiver 242 which receives data signals from satellite 244 which in turn is in remote communication with satellite signal transmitter 246. Terminals, for example further processing system 248, notebook computer 250 or satellite telephone 252, can thereby communicate with network 202. A local network 260, which for example may be a private network, LAN, etc., may also be connected to network 202. For example, network 202 could be connected with Ethernet 262 which connects terminals 264, server 266 which controls the transfer of data to and/or from database 268, and printer 270. Various other types of networks could be utilised.

[0062] The processing system 100 is adapted to communicate with other terminals, for example further processing systems 206, 208, by sending and receiving data, 118, 120, to and from the network 202, thereby facilitating possible communication with other components of the networked communications system 200. Thus, for example, the networks 202, 230, 240 may form part of, or be connected to, the Internet, in which case, the terminals 206, 212, 218, for example, may be web servers, Internet terminals or the like. The networks 202, 230, 240, 260 may be or form part of other communication networks, such as LAN, WAN, Ethernet, token ring, FDDI ring, star, etc., networks, or mobile telephone networks, such as GSM, CDMA or 3G, etc., networks, and may be wholly or partially wired,

including for example optical fibre, or wireless networks, depending on a particular implementation.

[0063] Accordingly, in the broad manner described, the processing systems 12.1 and control server 18 are generally realised by suitable versions of the broad processing system 100, as described above, and networked together to perform the functions and provide the features broadly described herein.

[0064] With reference now to Figure 3 of the accompanying drawings, the system 10 for synchronised telematic music collaboration generally comprises at least two MIDI-enabled user processing systems 12.1 that are arranged in signal communication with the internet 16, as well as the control server 18 also networked via the internet 16. As will become apparent, the system 10 may comprise a plurality of MIDI-enabled user processing systems 12.1, i.e. more than two, to allow a plurality of musicians and performers 12.2 to collaborate in real-time.

[0065] The remote server 18 is generally configured to, upon request, create a collaboration session on each user processing system 12.1, where each such session is linked to the control server 18 by assigned UDP and TCP ports which are allocated MIDI channels. As will be appreciated by the skilled addressee, such a session is typically virtually created in a memory arrangement of the user processing system 12.1, as described above, by means of local software providing a suitable user interface using suitable APIs, or the like.

[0066] In a preferred embodiment, the at least two MIDI-enabled user processing systems 12.1 and the control server 18 are arranged in signal communication, via the internet, by means of a star network topology with said control server 18

as a central hub. This approach has the advantage that there is no extra latency of data packets going from performer A 12.2 -> control server 18 -> performer B 12.2.

[0067] In one embodiment, the control server 18 creates the collaboration session by allocating MIDI port numbers to each session to coordinate MIDI sharing between sessions. In one embodiment, the control server 18 routes 16 channels of MIDI per IP port allocation, including 128 notes and 128 Control Change controllers per channel. For example, the control server 18 may be configured with software which enables 2 x IP ports to send/receive music performance data between two remote collaborators 12.2, as described in more detail below.

[0068] The remote server 18 is further configured to establish a UDP MIDI data stream with each session via said allocated MIDI channels, as well as establish a multiplexed TCP control MIDI data stream with each session. The control server is also configured to configure or adapt each session, i.e. each user processing system 12.1 via a suitable software interface, periodically to transmit a local timestamp signal based on Network Time Protocol (NTP), or the like.

[0069] The remote server 18 is further configured to exchange all MIDI data for the session via both allocated UDP and TCP ports, i.e. all MIDI data is respectively sent via both UDP and TCP ports and respective UDP and TCP MIDI data streams. In this manner, real-time music is shareable between sessions via both UDP and TCP MIDI data streams with the TCP control MIDI data stream useable to synchronise said shared UDP and TCP MIDI data streams via UDP and TCP MIDI data stream and periodic local timestamp signal comparison.

[0070] As UDP is more suited to real-time sharing of information due to minimum transmission latency, MIDI data is shareable between sessions in substantial real-time. Importantly, whereas TCP is more reliable and allows error checking, it does introduce latency, but TCP is useable to transmit data that isn't as time sensitive as real-time music, but is important to maintain the integrity and synchronisation of the shared collaboration session. This control stream is typically multiplexed so that only a single TCP stream is required but multiple types of data can be exchanged between participants.

[0071] The skilled addressee is specifically to appreciate that the present invention generally performs latency checks via TCP, rather than UDP. This is different to conventional approaches, where a latency check is typically based on the arrival time of the UDP packet. The present invention is concerned with the timestamp in the TCP data packet and from which the temporal difference between two processing systems, such the control server 18 and a user processing system 12.1, can be determined, thereby rendering the transmission delay of using TCP less important.

[0072] The present invention typically exchanges four timestamps (two in each direction) between processing systems to provide an accurate way of determining the difference between the clocks in the two processing systems, with such clock differences then used to perform clock skew and messaging ordering correction for MIDI messages. In particular, the MIDI data streams between sessions are shared via both UDP and TCP ports as respective UDP and TCP MIDI data streams. In effect, the TCP MIDI data stream acts as a ‘safety net’ for lost UDP MIDI data, rather than as a complete version of transferred MIDI data.

[0073] In one embodiment, the control server 18 maintains the integrity of the UDP MIDI data stream, via the TCP control MIDI data stream, by prioritising MIDI 'note off', `pitch bend' and `Controller Change' messages between sessions. As it is not uncommon the get 'stuck notes' with MIDI transmissions when the note-off message packet is lost, the currently playing music note will continue 'hanging', which can be minimised with the control server 18 prioritising MIDI 'note off' messages between sessions. Similarly, the same applies to `pitch bend' messages when a return to `center position' packet is lost, where a voice remains detuned, and to `Controller Change' messages, e.g. volume, where if packets are lost a critical divergence may occur between session nodes.

[0074] In one embodiment, the control server 18, generally via the TCP control MIDI data stream, maintains an integrity of the UDP MIDI data stream by determining latency between each session via the periodic local timestamp signals from each user processing system 12.1 and synchronising the UDP MIDI data streams accordingly. The control server 18 is thus configured to manage MIDI data send/receive functions, e.g. receive on one channel, send on a unified channel, ensure lost data packets are received by all parties via MIDI TCP, e.g. 'note off' MIDI messages, synchronising collaborator processing systems and devices 12.1 using shared UTC start-on-time commands, and monitor data synchronisation between all collaborators via periodic MIDI local timestamp signals.

[0075] For example, in one embodiment the protocol identifiers for the UDP MIDI data stream packet format may comprise:

Protocol identifier (1 octet)

Id 1: MIDI-compatible control data

Id 3: MIDI-compatible notes data

Id 16: AR/VR control data
Id 18: AR/VR session data
Id 32: Audio stream control data
Id 34: Audio stream data
Id 48: Video stream control data
Id 50: Video stream data.

[0076] Similarly, in one example the TCP control MIDI data stream packet format may comprise:
Id 0: Session setup and maintenance
Id 1: MIDI-compatible control data
Id 3: MIDI-compatible notes data
Id 16: AR/VR control data
Id 18: AR/VR session data
Id 32: Audio stream control data
Id 34: Audio stream data
Id 48: Video stream control data
Id 50: Video stream data
Id 100: Generic binary transfer stream data.
Of course, variations on the above are possible and anticipated.

[0077] In one embodiment, the control server 18 is configured to prepare and transmit a TDP MIDI audience data stream comprising a combination of all MIDI data streams from all sessions that have been time-corrected for latency and delay between respective sessions. In this manner, the control server 18 may generate an audio and/or visual feed for an online audience 20, which may comprise various further devices or processing systems 22, as indicated. In such a manner, an audience 20 may be enabled to listen and/or view a MIDI data feed which may include an immersive 3D visual design software aspect configured to provide visual aspects for remote viewing

by the audience 20, e.g. people can stream the collaboration session from the control server 18.

[0078] In a typical embodiment, each MIDI-enabled user processing system 12.1 comprises a suitable software application which is configured to provide a GUI for the collaboration session established by the control server. In one embodiment, each MIDI-enabled user processing system 12.1 is preconfigured, via such a suitable software application, with local audio files and/or synthesisers that are trigged by the UDP MIDI data stream and/or TCP control MIDI data stream. Of course, such suitable software application may take various forms, including a dedicated software application, a browser-enabled GUI, a direct network connection application, and/or the like.

[0079] In the manner described, the system 10 facilitates the avoidance of feedback loops between collaborators 12.2 and provides functionality for MIDI actions for beats, bass, harmony and melody, recording of performed loops in these four categories via preconfigured software with music loop play/stop functionality further to reduce bandwidth requirements between collaborators in a session, as well as the application of numerous Digital Signal Processing (DSP) actions, such as effects. System 10 also allows quantising music on/off functionality and future time start functions to synchronise sessions, with the periodic MIDI timestamp signals from each user processing system 12.1 to measure latency variation during a collaboration performance.

[0080] In one embodiment, the control server 18 may further be configured to encrypt the TCP control MIDI data stream. In order for firewalls and network routers to properly inspect traffic and open ports it may be necessary to start a session

unencrypted and then transition to using encryption for specific payloads. To this end, per-packet UDP-style encryption may be required with the TCP control MIDI data stream for multiplexed traffic where some is encrypted and other traffic is not.

[0081] The present disclosure further includes an associated method for synchronised telematic music collaboration. Such a method generally comprises the steps of:

by means of the control server 18, creating a collaboration session on at least two MIDI-enabled user processing systems 12.1 via the internet, each user processing system linked to the control server 18 by an assigned UDP and TCP port which is allocated MIDI channels, each user processing system configured periodically to transmit a local timestamp signal based on Network Time Protocol (NTP);

via the control server 18, establishing a UDP MIDI data stream with each session via the UDP port;

via the control server 18, establishing a multiplexed TCP control MIDI data stream with each session via the TCP port;

via the control server 18, exchange all MIDI data for the session via both allocated UDP and TCP ports; and

via the control server 18, synchronising all sessions by comparing both UDP and TCP MIDI data streams using said local timestamp signals.

[0082] In this manner, as described, real-time music is shareable between sessions via both UDP and TCP MIDI data streams with the TCP control MIDI data stream useable to synchronise said shared UDP and TCP MIDI data streams via UDP and TCP MIDI data stream and periodic local timestamp signal comparison.

[0083] Applicant believes it particularly advantageous that the present invention provides for system 10 and an associated method which allows for synchronised remote musical collaboration in real-time, which prioritises a rhythmic hierarchy of musical actions without introducing transmission latency by using MIDI UDP data less affected by network latency and jitter, able to be quantised, and able to generate high-definition audio upon arriving at each user processing device 12.1. The invention also allows MIDI data from performance actions to be harnessed to generate visual asset behaviour by the control server 18 generating an audio and/or visual feed for an online audience 20, e.g. an AR and/or VR immersive feed, or the like.

[0084] The MIDI data exchange protocol described herein is designed to be extensible by adding new message types - up to 256 message types are possible. An eight-bit message type number was chosen to give sufficient flexibility without introducing too much overhead on the network. MIDI data is sent as a particular message type on both TCP and UDP. The protocol has message types defined for sending other binary data on TCP (such as music/sound samples). In this case, sending via UDP is not necessary as samples or other non-time-sensitive performance data are likely to be sent before a performance.

[0085] An exception might be a loop construct created by a performer during a live event - due to the size of the data and the criticality it makes sense to send this via TCP due to guaranteed delivery and a lower use of bandwidth (as compared to sending on both UDP and TCP). Other time-sensitive performance message types would be sent on both UDP and TCP for the same reason MIDI messages use UDP and TCP. An example is motion capture data. Much like MIDI does with audio, rather

than capture a video of performers and sending a frame of pixels the motion capture system only sends the motion data, greatly reducing bandwidth but relying on the remote system to reconstruct the scene using that data. This type of data is ideal for the protocol of the present invention and is easily added by creating a new message type, for example for motion capture data. If it is time critical/sensitive, as it generally is in the case of motion capture data, then this data is sent via UDP and TCP in the same manner as MIDI messages are.

[0086] The protocol of the present invention may also find further applications in, for example, online computer gaming where time-sensitive data needs to be delivered reliably to multiple participants - in some cases ensuring that data does not arrive early for those players who have a lower latency connection to the server than others.

[0087] Optional embodiments of the present invention may also be said to broadly consist in the parts, elements and features referred to or indicated herein, individually or collectively, in any or all combinations of two or more of the parts, elements or features, and wherein specific integers are mentioned herein which have known equivalents in the art to which the invention relates, such known equivalents are deemed to be incorporated herein as if individually set forth. In the example embodiments, well-known processes, well-known device structures, and well-known technologies are not described in detail, as such will be readily understood by the skilled addressee.

[0088] The use of the terms "a", "an", "said", "the", and/or similar referents in the context of describing various embodiments (especially in the context of the claimed subject

matter) are to be construed to cover both the singular and the plural, unless otherwise indicated herein or clearly contradicted by context. The terms "comprising," "having," "including," and "containing" are to be construed as open-ended terms (i.e., meaning "including, but not limited to,") unless otherwise noted. As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items. No language in the specification should be construed as indicating any non-claimed subject matter as essential to the practice of the claimed subject matter.

[0089] It is to be appreciated that reference to "one example" or "an example" of the invention, or similar exemplary language (e.g., "such as") herein, is not made in an exclusive sense. These examples are intended to assist the skilled person in performing the invention and are not intended to limit the overall scope of the invention in any way unless the context clearly indicates otherwise. Variations (e.g. modifications and/or enhancements) of one or more embodiments described herein might become apparent to those of ordinary skill in the art upon reading this application. The inventor(s) expects skilled artisans to employ such variations as appropriate, and the inventor(s) intends for the claimed subject matter to be practiced other than as specifically described herein.

[0090] Any method steps, processes, and operations described herein are not to be construed as necessarily requiring their performance in the particular order discussed or illustrated, unless specifically identified as an order of performance. It is also to be understood that additional or alternative steps may be employed.

CLAIMS

1. A system for synchronised telematic music collaboration comprising:

at least two MIDI-enabled user processing systems arranged in signal communication with the internet; and

a control server arranged in signal communication with the internet, said remote server configured to:

i. upon request, create a shared collaboration session on each user processing system, each session linked to the control server by assigned UDP and TCP ports allocated to the collaboration session;

ii. establish a UDP MIDI data stream with each session via said allocated UDP port;

iii. establish a multiplexed TCP control MIDI data stream with each session via said allocated TCP port;

iv. configure each session periodically to transmit a local timestamp signal based on the Network Time Protocol (NTP); and

v. exchange all MIDI data for the session via both allocated UDP and TCP ports;

wherein real-time music is shareable between sessions via both UDP and TCP MIDI data streams with the TCP control MIDI data stream useable to synchronise said shared UDP and TCP MIDI data streams via UDP and TCP MIDI data stream and periodic local timestamp signal comparison.

2. The system of claim 1, wherein the control server, via the TCP control MIDI data stream, synchronises the shared MIDI data streams by determining latency between each session via the periodic local timestamp signals from each session and

calibrating exchange of MIDI data between sessions accordingly.

3. The system of either of claims 1 or 2, wherein the control server, via the TCP control MIDI data stream, maintains an integrity and synchronisation of the UDP and TCP MIDI data streams by comparing the TCP MIDI data stream to the UDP MIDI data stream to identify missing or delayed MIDI UDP data.

4. The system of any of claims 1 to 3, wherein the at least two MIDI-enabled user processing systems and the control server are arranged in signal communication, via the internet, by means of a star network topology with said control server as a central hub.

5. The system of any of claims 1 to 4, wherein the control server creates the collaboration session by allocating UDP and TCP port numbers to each session to coordinate MIDI sharing between sessions.

6. The system of any of claims 1 to 5, wherein the control server routes 16 channels of MIDI per UDP and TCP port allocation, including 128 notes and 128 Control Change controllers per channel and additional MIDI signals as per the RFC 6295 RTP Payload Format for MIDI specification.

7. The system of any of claims 1 to 6, wherein the control server maintains the integrity of the UDP MIDI data stream, via the TCP control MIDI data stream, by prioritising MIDI ‘note off’, ‘pitch bend’ and ‘Controller Change’ messages between sessions.

8. The system of any of claims 1 to 7, wherein the control server is configured to generate an optional audio and/or visual feed for an online audience.

9. The system of any of claims 1 to 8, wherein the MIDI-enabled user processing system comprises a MIDI musical instrument.

10. The system of any of claims 1 to 9, wherein each MIDI-enabled user processing system comprises a software application configured to provide a GUI for the collaboration session established by the control server.

11. The system of any of claims 1 to 10, wherein the MIDI-enabled user processing system is preconfigured, via a software application, with local audio files and/or synthesisers that are trigged by the UDP MIDI data stream and/or TCP control MIDI data stream.

12. The system of any of claims 1 to 11, wherein the control server is configured to encrypt the TCP control MIDI data stream.

13. The system of any of claims 1 to 12, wherein the control server is configured to prepare and transmit a TDP MIDI audience data stream comprising a combination of all MIDI data streams from all sessions that have been time-corrected for latency and delay between respective sessions.

14. A method for synchronised telematic music collaboration, said method comprising the steps of:

by means of a control server, creating a collaboration session on at least two MIDI-enabled user processing systems via the internet, each user processing system linked to the

control server by an assigned UDP and TCP port which is allocated MIDI channels, each user processing system configured periodically to transmit a local timestamp signal based on Network Time Protocol (NTP);

via the control server, establishing a UDP MIDI data stream with each session via the UDP port;

via the control server, establishing a multiplexed TCP control MIDI data stream with each session via the TCP port;

via the control server, exchange all MIDI data for the session via both allocated UDP and TCP ports; and

via the control server, synchronising all sessions by comparing both UDP and TCP MIDI data streams using said local timestamp signals;

wherein real-time music is shareable between sessions via both UDP and TCP MIDI data streams with the TCP control MIDI data stream useable to synchronise said shared UDP and TCP MIDI data streams via UDP and TCP MIDI data stream and periodic local timestamp signal comparison.

15. The method of claim 14, wherein the step of synchronising the UDP and TCP MIDI data streams comprises the control server determining latency between each session via the periodic local timestamp signals from each session and calibrating exchange of MIDI data between sessions accordingly.

16. The method of either of claims 14 or 15, wherein the step of synchronising the UDP and TCP MIDI data streams comprises the control server comparing the TCP MIDI data stream to the UDP MIDI data stream to identify missing or delayed MIDI UDP data.

17. The method of any of claims 14 to 16, wherein the step of creating a collaboration session between the at least two MIDI-enabled user processing systems and the control server

comprises arranging the MIDI-enabled user processing systems and the control server in signal communication, via the internet, by means of a star network topology with said control server as a central hub.

18. The method of any of claims 14 to 17, wherein the step of creating the collaboration session via the control server comprises said control server allocating MIDI port numbers to each session to coordinate MIDI sharing between sessions.

19. The method of any of claims 14 to 18, wherein the step of creating the collaboration session via the control server comprises the control server routing 16 channels of MIDI per UDP and TCP port allocation, including 128 notes and 128 Control Change controllers per channel and additional MIDI signals as per the RFC 6295 RTP Payload Format for MIDI specification.

20. The method of any of claims 14 to 19, wherein the control server maintains the integrity of the UDP MIDI data stream, via the TCP control MIDI data stream, by prioritising MIDI 'note off', 'pitch bend' and 'Controller Change' messages between sessions.

21. The method of any of claims 14 to 20, which includes a step of generating an audio and/or visual feed for an online audience via the control server.

22. The method of any of claims 14 to 21, wherein the step of creating a collaboration session on each user MIDI-enabled processing system comprises installing a software application on such processing system, said software application configured to provide a GUI for the collaboration session.

23. The method of any of claims 14 to 22, wherein the MIDI-enabled user processing system is preconfigured, via a software application, with local audio files and/or synthesisers that are trigged by the UDP MIDI data stream and/or TCP control MIDI data stream.

24. The method of any of claims 14 to 23, which includes a step of encrypting the TCP control MIDI data stream via the control server.

25. The method of any of claims 14 to 24, which comprises the step of, via the control server, preparing and transmitting a TDP MIDI audience data stream comprising a combination of all MIDI data streams from all sessions that have been time-corrected for latency and delay between respective sessions.

26. A computer programme product which, when executed by a suitable processing system, facilitates the performance of the method according to any of claims 14 to 25.