

Metodo per il recupero in tempo reale degli errori di trasmissione di segnali audio in formato MIDI

Original

Metodo per il recupero in tempo reale degli errori di trasmissione di segnali audio in formato MIDI / Rottondi, CRISTINA EMMA MARGHERITA; Cuccarese, Antonio; Bianco, Andrea. - (2020).

Availability:

This version is available at: 11583/2974100 since: 2022-12-22T20:03:02Z

Publisher:

Published

DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Metodo per il recupero in tempo reale degli errori di trasmissione di segnali audio in formato MIDI

DESCRIZIONE

5 **CAMPO TECNICO**

La presente invenzione si riferisce ad un metodo per la correzione degli errori dovuti a perdite o ritardi di pacchetti contenenti segnali audio in formato simbolico differenziale (quale la codifica MIDI) scambiati tra due o più musicisti collegati ad una rete di telecomunicazioni per applicazioni di Networked Music
10 Performance.

STATO DELL'ARTE

Un formato musicale simbolico differenziale, quale ad esempio quello definito nella la codifica Musical Instrument Digital Interface (MIDI), può essere generato e ricevuto da diversi dispositivi, come riassunto a titolo di esempio in
15 Figura 1: uno strumento musicale elettronico, anche detto strumento MIDI, uno strumento musicale tradizionale di tipo analogico collegato ad un convertitore MIDI, un dispositivo elettronico/computer in grado di generare/ricevere un flusso MIDI, anche di controllo. In Figura 1 è anche riportato un blocco denominato MIDI extender, oggetto del metodo di questo brevetto, che può
20 essere realizzato in hardware (HW) o in software (SW), il cui funzionamento sarà descritto in seguito. Lo standard MIDI permette la generazione di dati digitali come rappresentazione di note musicali da riprodurre e/o comandi di controllo che regolano il tipo di suono in riproduzione e che possono essere trasmessi ad

altri dispositivi, e/o essere utilizzati da applicazioni software per la registrazione, la riproduzione o la modifica dei dati digitali stessi.

Lo standard MIDI definisce quindi un insieme di messaggi per la comunicazione tra dispositivi. Un flusso di messaggi MIDI può essere trasmesso da un dispositivo *mittente* ad un (o più) altro dispositivo *ricevente*, attraverso un apposito cavo MIDI, che collega direttamente i dispositivi, ad esempio uno strumento musicale ed una scheda audio. In alternativa, i dati tra i dispositivi possono essere trasmessi da un dispositivo mittente attraverso una connessione di rete dati, come ad esempio una rete wireless di corto raggio come una rete WiFi oppure una rete di telecomunicazioni di tipo geografico come la rete Internet, mediante l'invio di messaggi MIDI incapsulati in pacchetti dati.

In una rete dati, il requisito primario per la maggior parte delle applicazioni di trasferimento dati è l'affidabilità della trasmissione dei pacchetti, ovvero la probabilità che i pacchetti siano ricevuti correttamente. Con particolare riferimento a pacchetti contenenti messaggi MIDI, per mantenere una buona qualità sonora, è importante gestire eventuali perdite di pacchetti dovute ad esempio alla congestione della rete. Per altre applicazioni, ad esempio quelle multimediali con requisiti di tempo reale, assume invece importanza fondamentale anche la latenza, ovvero l'intervallo di tempo che intercorre tra l'invio di un pacchetto di dati da parte del mittente e la sua ricezione da parte del ricevitore. La bassa latenza diventa fondamentale, soprattutto per sessioni musicali in tempo reale, in applicazioni denominate network music performance (NMP).

Per applicazioni dati tradizionali oppure applicazioni multimediali in

streaming senza requisiti di tempo reale, per ridurre la perdita di dati in rete si predilige l'utilizzo di un protocollo di livello trasporto affidabile come il TCP (Transmission Control Protocol). Questo protocollo ritrasmette dopo un certo tempo i dati persi in rete o mancanti, introducendo quindi un aumento della
5 latenza oltre che del traffico sulla rete.

Quando un flusso di dati MIDI generato da un musicista viene trasmesso ad un dispositivo remoto di ricezione per riprodurre il suono, un ritardo di trasmissione elevato può essere notato da un ascoltatore, da altri musicisti che collaborano alla performance e da un eventuale pubblico. Pertanto, per
10 applicazioni sonore con requisiti di tempo reale quali quelle di NMP, i messaggi MIDI possono essere trasmessi in rete con protocolli inaffidabili come l'UDP (User Datagram Protocol) per evitare i meccanismi di ritrasmissione del protocollo TCP. Questa scelta evita l'aumento della latenza indotto dalla ritrasmissione di dati dei protocolli di trasporto affidabili, ma non prevede alcuna
15 azione da parte del protocollo UDP in caso di perdita dei dati.

Si consideri una sessione musicale in rete tra un gruppo di musicisti che interagiscono tra loro pur trovandosi in luoghi fisici diversi, per suonare insieme come se si trovassero nello stesso luogo. L'introduzione di intervalli di silenzio o di ritardi temporali può compromettere la qualità della sessione musicale,
20 facendo perdere ai musicisti il ritmo o l'armonia. In questi casi, le perdite di dati vanno minimizzate, o, meglio ancora eliminate, e le latenze devono essere le più ridotte possibili per garantire una interazione musicale naturale.

Nello stato dell'arte sono note numerose tecniche che cercano di mitigare gli errori di trasmissione attraverso una connessione di rete inaffidabile.

Ad esempio, con riferimento a una rete wireless fra terminali mobili e.g. telefoni cellulari, nel brevetto US 6,898,729 i dati MIDI vengono analizzati e classificati in due categorie (critica e non critica) e successivamente trasmessi rispettivamente attraverso una connessione affidabile ed una inaffidabile.

5 Tipicamente i comandi MIDI che segnalano quando una determinata nota vada suonata e quando fermata vengono trasmessi per mezzo della connessione affidabile. Questa soluzione tuttavia implica l'utilizzo di due connessioni i.e. affidabile e inaffidabile per la trasmissione dei dati.

Nel brevetto US 7,447,639, che fa riferimento a dati inviati in streaming,
10 ovvero non in tempo reale, viene illustrato un sistema di occultamento degli errori, lato dispositivo ricevente, attraverso l'estrazione dei pattern ritmici dei segnali musicali incorporati come dati ausiliari lato dispositivo trasmittente. Questa soluzione comporta l'estrazione e l'analisi del pattern ritmico ed è particolarmente adatta per la sola riproduzione precisa di suoni, ad esempio
15 quando viene trasmessa musica in streaming in cui sono presenti pattern ritmici di batteria complessi e veloci. Non viene considerata l'esigenza di gestire sorgenti sonore MIDI con bassa latenza. Al contrario, l'analisi del pattern ritmico è un'operazione complessa che richiede tempi relativamente elevati e non si adatta particolarmente allo scambio e mixaggio con bassa latenza di tracce sonore
20 generate da due o più utenti collegati alla rete.

Nel brevetto US 9,601,097 B2, con riferimento alla trasmissione dati tramite WiFi fra strumenti musicali MIDI e altri dispositivi MIDI, viene descritto un nuovo protocollo basato su UDP che prevede la ritrasmissione dei dati non ricevuti, implementando una memoria utilizzata per la storicizzazione dei dati

inviati. Ogni ricezione di un pacchetto dati innesca una trasmissione di conferma ricezione. La memoria diventa il riferimento per la ritrasmissione dell'intero storico dati precedentemente trasmessi e non confermati. Questa soluzione comporta un meccanismo di riscontro dei pacchetti inviati con la relativa
5 generazione di traffico legata a conferme e ritrasmissioni. Questo meccanismo aumenta la latenza di ricezione del suono e non è quindi adatta alla trasmissione di segnali MIDI in tempo reale per applicazioni di NMP.

La domanda di brevetto US 2004/0154460, con riferimento a reti di scambio dati a radiofrequenza a corto raggio, descrive un sistema per la sostituzione di
10 informazioni mancanti o corrotte al fine di evitare gli intervalli di silenzio. Questa soluzione, dopo aver valutato la severità dell'errore, attiva una fase di recupero dei dati per mezzo di un canale ausiliario asincrono attraverso un server. La soluzione presuppone quindi sia l'utilizzo di un canale ausiliario che l'utilizzo di un server ed essendo basata su un meccanismo di ritrasmissione aumenta la
15 latenza del segnale sonoro.

SCOPI E RIASSUNTO DELL'INVENZIONE

Lo scopo della presente invenzione è quello di risolvere almeno parzialmente gli svantaggi legati alla trasmissione dei dati MIDI o di altro protocollo simbolico differenziale tramite una rete di scambio dati evitando i) l'introduzione di sistemi
20 di conferma concettualmente simili ai meccanismi tipici delle connessioni affidabili e ii) la replica dei segnali ricevuti in precedenza, le tecniche tradizionalmente utilizzate per di evitare silenzi indesiderati o alterazioni, mantenendo una bassa latenza, come richiesto per applicazioni di NMP, ovvero per consentire una riproduzione in tempo reale dei suoni musicali scambiati fra

due o più utenti della rete. A tale scopo la presente invenzione prevede l'utilizzo di un MIDI extender HW o SW (si veda **Errore. L'origine riferimento non è stata trovata.**) al trasmettitore, che si colloca tra il dispositivo musicale sorgente di messaggi MIDI ed il dispositivo in grado di accedere ad una rete di scambio dati per l'invio dei messaggi MIDI all'interno dei pacchetti dati. Il MIDI extender al 5 trasmettitore intercetta i messaggi MIDI provenienti da una sorgente MIDI (sia essa uno strumento musicale elettronico o analogico sensorizzato, un computer, etc.) e li incapsula in pacchetti secondo un determinato formato (descritto successivamente a titolo di esempio) aggiungendo periodicamente ai messaggi MIDI ricevuti messaggi di refresh che contengono l'insieme dei valori assunti dalle variabili (ad esempio le note attive), in continuo aggiornamento, che rappresentano le note e/o i comandi in esecuzione al tempo T per un dato dispositivo MIDI. Da ora in avanti, si farà riferimento al MIDI extender in trasmissione con il termine *sender*, ed al MIDI extender in ricezione con il termine 10 *receiver*. L'insieme dei valori delle variabili sopra menzionato definisce lo stato di un dispositivo MIDI lato sender.

Secondo la presente invenzione, lo scopo viene raggiunto tramite un metodo di comunicazione di scambio dati rappresentativi di note musicali da riprodurre e/o comandi di controllo che regolano il suono in riproduzione, realizzato nel MIDI extender, comprendente le seguenti fasi in trasmissione (e quelle duali in ricezione):

ricevere un flusso di dati rappresentativo di suoni di uno strumento musicale tramite una pluralità di messaggi di dati formattati secondo un formato differenziale simbolico;

aggiornare ripetitivamente i valori di una pluralità di variabili codificate in detti messaggi;

memorizzare, per una pluralità di variabili, una relativa lista di valori definente uno stato di uno strumento musicale in istanti di tempo successivi e
5 progressivi;

generare pacchetti di dati comprendenti una marca temporale e una pluralità di detti messaggi relativi ad un dato intervallo di tempo associato alla marca temporale;

includere selettivamente nei pacchetti la detta lista per generare pacchetti
10 di refresh, ciascuno riferito a uno stato dello strumento musicale in uno dei detti istanti di tempo;

trasmettere i pacchetti, includendo i pacchetti di refresh, in una rete di scambio dati.

La presente invenzione utilizza meccanismi di aggiornamento dei dati
15 generati da un dispositivo musicale e/o computer collegati al sender e trasmessi al receiver, per permetterne la riproduzione in tempo reale tramite e.g. un altoparlante, una cuffia o altro dispositivo atto allo scopo. In particolare, i pacchetti inviati contengono selettivamente, in aggiunta ai messaggi MIDI, i valori assunti dalle variabili di processo e/o stato del dispositivo MIDI relativi a
20 note o comandi basati su detti messaggi all'istante T del sender. Questo permette al receiver di correggere l'insieme dei valori delle proprie variabili relative alle note e/o comandi di controllo in esecuzione sul sender al tempo T e quindi, da un lato di ottenere un'interazione musicale più naturale tra musicisti poiché

viene evitata la ritrasmissione di dati, e dall'altro di migliorare significativamente la qualità audio riprodotta attraverso un modesto incremento dei dati trasmessi tra sender e receiver. L'invio dei pacchetti di refresh può avvenire anche in modo adattativo in funzione della congestione della rete e della percentuale dei pacchetti persi. Con particolare riferimento ai protocolli MIDI come MIDI 2.0, MIDI 1.0 (ulteriori informazioni sono reperibili presso il sito www.midi.org), i messaggi contenenti note o comandi di controllo che variano i valori delle variabili di un dispositivo MIDI possono essere ad esempio Note ON, Note OFF, System Exclusive, Program Change etc. In modo dipendente dalle forme di realizzazione, i dati organizzati in pacchetti aventi una marca temporale, una pluralità di messaggi ed una pluralità di valori di variabili, sono generati in modo nativo a bordo dello strumento musicale MIDI che integra il MIDI extender oppure da un dispositivo separato dallo strumento musicale MIDI e ricevente in ingresso dati relativi a suoni generati dallo strumento.

Secondo una forma preferita di realizzazione, un MIDI extender in ricezione, che può essere un dispositivo elettronico programmabile oppure un personal computer oppure un telefono cellulare oppure un tablet, riceve i pacchetti di dati inclusi i pacchetti di 'refresh' e genera un segnale di ingresso per un dispositivo di emissione acustica come un altoparlante oppure una cuffia sulla base dell'insieme dei valori delle variabili che definiscono lo stato dello strumento musicale che genera i dati d'ingresso al MIDI extender. In questo modo, se a causa di un errore di trasmissione un messaggio Note OFF non raggiunge il dispositivo ricevente perché il relativo pacchetto è stato ricevuto in ritardo o è andato perso generando così un suono indesideratamente prolungato,

il pacchetto di refresh contiene le opportune informazioni che permetteranno al MIDI extender di ricezione di allineare il valore delle variabili presenti sul sender nell'istante temporale a cui si riferisce il pacchetto di refresh ricevuto e dunque terminare la riproduzione della nota.

5 Secondo una forma preferita di realizzazione, allo scopo di ridurre la quantità scambiata di dati, i pacchetti di refresh contengono i valori delle variabili in istanti di tempo progressivi e definenti intervalli di tempo la cui ampiezza è costante o variabile. Tale ampiezza è maggiore, ad esempio un multiplo intero, dell'ampiezza degli intervalli di tempo relativamente ai quali sono generati i
10 pacchetti che incapsulano i messaggi MIDI.

Inoltre, la presente invenzione prevede di regolare in modo adattivo gli intervalli di tempo ai quali si riferiscono i dati di stato contenuti nei pacchetti di refresh sulla base della capacità della rete di scambio dati.

BREVE DESCRIZIONE DEI DISEGNI

- 15 - La Figura 2 mostra un esempio di interazione fra tre musicisti su cui è evidenziato solo il traffico del musicista 1.
- La Figura 3 mostra lo schema della struttura di un pacchetto MIDI generato dalla presente invenzione.
- La Figura. 4 mostra il dettaglio della sezione B della struttura di un
20 pacchetto MIDI.
- La Figura 5 mostra il dettaglio della sezione C della struttura di un pacchetto MIDI.
- La Figura 6 mostra il diagramma a blocchi della routine di controllo effettuata dal receiver all'arrivo di ogni nuovo pacchetto dati.

- La Figura 7 mostra l'effetto della frequenza di invio dei pacchetti di refresh al fine dell'accuratezza con cui avviene la ricostruzione del segnale MIDI originale per migliorare la qualità dell'esecuzione.

DESCRIZIONE DETTAGLIATA DELL'INVENZIONE

5 La presente invenzione è applicabile a un sistema di comunicazione (Figura 2) preferibilmente per NMP con almeno due musicisti collegati coi relativi strumenti musicali 1 comprendente, per ciascuno strumento musicale, un dispositivo elettronico di controllo preferibilmente a bordo dello strumento per generare segnali audio secondo un formato simbolico differenziale sulla base
10 dell'azione del musicista sullo strumento musicale digitale. Il sistema comprende inoltre un MIDI extender, denominato primo computer PC1, o altro controllore programmabile, collegato con una rete di scambio dati, in particolare una rete Internet di scambio dati, che riceve un flusso di dati, preferibilmente messaggi codificati tramite il protocollo MIDI, dallo strumento musicale digitale e li
15 elabora in pacchetti prima di trasmetterli a un secondo MIDI extender, denominato computer PC2, collegato alla rete e programmato per ricevere ed elaborare i pacchetti di dati e riprodurre il suono dello strumento musicale digitale PC1 tramite un altoparlante oppure cuffie o simili. Allo scopo di realizzare una NMP, anche PC2 è collegato ad uno strumento musicale digitale
20 ed elabora i relativi dati per inviarli tramite la rete al computer PC1, che li elabora per riprodurli tramite un proprio altoparlante. Pertanto entrambi i computer PC1 e PC2 agiscono nella rete come sender e receiver di dati MIDI.

Secondo l'invenzione, pacchetti di dati codificati tramite il formato audio simbolico differenziale (MIDI) sono integrati con una rappresentazione assoluta

(e non differenziale) in una pluralità di istanti di tempo noti e monitorati dell'insieme dei valori delle variabili di ogni sender ciascuno dei quali si può riferire ad una nota e/o ad un comando, e.g. quale nota viene generata dallo strumento musicale oppure, nel caso di un controller MIDI collegato alla rete internet di scambio dati, quali effetti sono applicate dal controller ai suoni degli
5 strumenti musicali collegati ai PC1 e PC2. In questo modo, se un comando Note OFF eseguito sullo strumento musicale 1 non viene ricevuto per errore dal ricevitore PC2, quando il ricevitore PC2 riceverà la lista dei valori delle variabili di stato dello strumento musicale 1 ad un tempo assoluto T inviata da PC1, potrà
10 aggiornare lo stato delle proprie variabili relative alle note e/o i comandi di controllo in esecuzione su PC1 e, in particolare, potrà inferire che il comando NOTE OFF è stato perso e procedere di conseguenza i.e. interrompendo la relativa nota. Per messaggio simbolico differenziale si intende un'istruzione di cambiamento di stato relativo a un parametro di codifica del segnale sonoro dello
15 strumento musicale 1. Ad esempio il messaggio "Note ON" comprende, opportunamente codificati in byte: il comando stesso "Note ON" codificato ed il canale su cui va applicato, una nota, un parametro della nota e.g. un volume o intensità. Tale comando consente al dispositivo elettronico ricevente e.g. PC2 con il relativo altoparlante, di iniziare la riproduzione del segnale acustico
20 corrispondente alla nota 'Note' specificata nel messaggio stesso, mentre un messaggio "Note OFF" farà terminare la suddetta riproduzione.

Nel presente documento sono usati i termini generici sender e receiver perché interessa una connotazione agnostica della rete e dei suoi nodi. Infatti l'invenzione qui descritta può concettualmente adattarsi, a titolo di esempio non

limitativo, sia ad applicazioni di tipo client/server, dove determinati nodi (client) richiedono un servizio a cui altri nodi (server) provvedono, sia ad applicazioni di tipo peer-to-peer, dove i nodi non sono individuati come client e server ma sotto forma di nodi operativamente equivalenti in grado di svolgere entrambe le
5 funzioni di invio e ricezione nei confronti di un altro nodo della rete. L'invenzione è inoltre applicabile a reti di scambio dati a corto raggio e wireless e.g. WIFI e a reti di telecomunicazione quali Internet, sia wireless, ad esempio tramite protocolli 4G/5G per dispositivi cellulari, sia via cavo, ad esempio fibre ottiche.

10 Si definiscano come parametri operativi due intervalli di tempo Δt_1 e Δt_2 , relativi alla periodicità con cui si inviano rispettivamente i messaggi MIDI e la rappresentazione assoluta del valore delle variabili che descrivono le note e/o i comandi presenti lato sender. Per semplicità di descrizione, si ipotizzi Δt_2 multiplo intero di Δt_1 . Il calcolatore/dispositivo elettronico a cui è collegato in
15 scambio dati lo strumento suonato dal musicista esegue un'applicazione MIDI di NMP dedicata, la quale comunica, utilizzando ad esempio il protocollo di trasporto UDP, con il calcolatore remoto che esegue l'applicazione MIDI di NMP ricevente i dati di suono e.g. pacchetti di messaggi MIDI.

20 La trasmissione dell'informazione tramite la rete è periodica, di periodo Δt_1 , ed è caratterizzata dall'invio da parte del MIDI extender di una pluralità di messaggi MIDI, raggruppati in un pacchetto (un esempio di una possibile struttura di tale pacchetto è descritta nel seguito e illustrata nelle figure 2-4). Ogni pacchetto, inviato alla fine di ciascun periodo di durata Δt_1 , contiene l'insieme dei messaggi MIDI generati sulla base dell'azione applicata allo strumento dal

musicista durante tale periodo di tempo. Secondo l'invenzione, alcuni pacchetti trasmessi da un sender comprendono l'insieme dei valori delle variabili che, con una periodicità Δt_2 , descrivono lo stato delle note e/o dei comandi di controllo relativi all'azione del musicista sullo strumento musicale 1 al tempo T i.e. pacchetti di refresh; Δt_2 identifica la frequenza o l'intervallo temporale con cui i valori delle variabili al tempo T vengono incorporate e.g. accodate all'interno del pacchetto di refresh. Per determinare i valori delle variabili presenti sul sender al tempo T , ogni sender esegue successive "fotografie" al tempo T del valore delle proprie variabili, che rappresentano le note e/o i comandi di controllo rappresentativi dell'azione del musicista sullo strumento musicale 1 al tempo T , effettuando una copia dell'insieme dei valori delle variabili per ciascun successivo intervallo di tempo Δt_2 . Secondo una forma di realizzazione preferita, i successivi istanti di tempo T in cui sono generati pacchetti di refresh sono definiti tramite un intervallo di tempo costante e quindi $T=0$, $T= \Delta t_2$, $T=2 \Delta t_2$, $T=3 \Delta t_2$ etc. E' inoltre importante notare che la marca temporale di ciascun pacchetto, inclusi quella nei pacchetti di refresh, unitamente agli *offset* dei messaggi MIDI permettono di sincronizzare i pacchetti di più utenti. In generale, ciascun istante di tempo T cade all'interno del relativo pacchetto di refresh opportunamente sincronizzato.

Pertanto, considerando $T=0$ il momento in cui inizia la NMP, la somma di tutti gli intervalli temporali Δt_2 restituisce il tempo T corrente. Gli intervalli temporali Δt_2 possono essere costanti durante la NMP e/o regolabili prima di iniziare la NMP e/o adattivi durante la NMP.

Nell'istante T , ogni sender verifica o determina il valore corrente delle

variabili (ad esempio una nota in riproduzione può aver ricevuto il comando di terminazione oppure viceversa può essere arrivato dall'interfaccia MIDI un comando per iniziare la riproduzione di un'altra nota) e questa informazione è quella che verrà inserita all'interno del pacchetto di refresh. In fase di ricezione da parte del receiver, questi pacchetti di refresh comprendenti le informazioni aggiuntive relative ai valori delle variabili permettono all'applicazione / controllore MIDI ricevente di riallineare periodicamente o a intervalli di tempo monitorati i valori delle proprie variabili che definiranno di conseguenza i comandi che sono eseguiti per riprodurre il segnale audio tramite l'altoparlante dal receiver, fungendo così da correzione di errori causati da perdite di pacchetti e quindi ripristinando la situazione corretta del sender al tempo T. Con riferimento all'esempio descritto in precedenza, dove non viene ricevuto (o non viene ricevuto per tempo) un pacchetto contenente il messaggio "Note ON" relativo ad una nota, senza questo meccanismo di riallineamento delle variabili basato sulle informazioni aggiuntive relative alle note e/o comandi di controllo rappresentativi dell'azione del musicista sullo strumento musicale 1 al tempo T e presenti sul relativo sender, il generico receiver X non avrebbe più modo di iniziare la riproduzione della nota se non potesse verificare attraverso la rappresentazione dei valori delle variabili di stato del sender che un determinato messaggio era stato generato in precedenza.

Secondo una prima forma preferita dell'invenzione quando il receiver riceve un pacchetto contenente una lista di valori di variabili di stato, i.e. un pacchetto di refresh, questa lista contiene messaggi MIDI che permettono direttamente il riallineamento delle note e/o comandi MIDI sul receiver. Secondo un'altra forma

preferita dell'invenzione al receiver verranno inviati unicamente dati rappresentativi dei valori delle variabili, eventualmente non in formato MIDI, ed è compito del MIDI extender del receiver, tramite opportuna programmazione, trasformare tali dati in messaggi MIDI che permettano il riallineamento fra le

5 note e/o i comandi di controllo presenti lato sender e quelli lato receiver. La rappresentazione più efficace di questa lista, anche in considerazione della quantità delle informazioni aggiuntive che si vanno ad aggiungere alla dimensione totale del pacchetto inviato, prevede che siano inserite quelle

10 variabili che rappresentano note e/o comandi di controllo ancora attivi all'istante T, cioè le variabili o i comandi MIDI che causano l'emissione di un suono tramite l'altoparlante del receiver. Questo non vuole essere limitativo nella considerazione di altre tipologie di rappresentazioni che portino al medesimo risultato come potrebbe essere un approccio duale a quello appena descritto: indicare tutte le variabili che non hanno subito un cambiamento nell'istante

15 considerato, a partire dai quali ottenere le note e/o i comandi attivi. Il valore assunto da Δt_2 può essere modificato durante la trasmissione dei dati, in funzione delle condizioni di congestione della rete e delle limitazioni di banda di trasmissione imposte dall'infrastruttura di rete stessa. Lo specifico criterio adottato per definire il valore di Δt_2 dipende inoltre dalla frazione di errori per

20 unità di tempo che si ritiene tollerabile per il musicista, ossia il rapporto tra il numero di messaggi musicali correttamente riprodotti lato ricevitore e il numero totale di messaggi generati dalla sorgente audio, misurati durante un dato intervallo di tempo.

Esempio di realizzazione pratica dell'invenzione

A mero titolo di esempio si descrive di seguito una possibile implementazione dell'invenzione con riferimento allo standard audio MIDI per dare maggiore chiarezza del funzionamento dell'invenzione.

Formato dei dati

5 Il messaggio MIDI è rappresentato come composizione di tre bytes:

- Il primo byte, chiamato Status byte, fornisce informazioni sul tipo di messaggio e sul canale MIDI cui si riferiscono.
 - Il secondo e il terzo byte, chiamati Data bytes, assumono diversi valori, il cui significato varia in base al tipo di messaggio specificato nello Status
- 10 byte.

Due istanze dell'applicazione NMP eseguite su host remoti scambiano messaggi MIDI aderendo, ad esempio, al seguente formato di pacchetto di livello applicativo (a cui ci si è riferiti nei paragrafi precedenti e si farà riferimento semplicemente come "pacchetto"). Il pacchetto è costituito da una stringa che

15 comprende diversi campi. La Figura 3 mostra una schematizzazione della struttura del pacchetto in cui sono visibili le tre sezioni che lo compongono:

- Una SEZIONE A contiene il *sequence number*, che indica univocamente il numero del pacchetto, e il *timestamp*, che indica il tempo assoluto al sender (in formato UTC) nel momento in cui il pacchetto è inviato. In generale è possibile

20 impiegare altri sistemi di marcatura temporale o sequenziale in modo che il receiver sia in grado di ordinare i pacchetti ricevuti dai vari sender.

- Una SEZIONE B contiene la sequenza di messaggi MIDI generati dallo strumento suonato dal musicista durante l'ultima finestra temporale di durata

Δt_1 . Come mostrato in Figura 4 ciascun messaggio è caratterizzato da un campo offset, uno sfasamento temporale calcolato rispetto all'istante di inizio della finestra corrente. Ad esempio, un messaggio con offset 5ms verrà riprodotto 5ms dopo l'inizio della riproduzione dei messaggi MIDI racchiusi nel pacchetto
5 stesso. Seguono 3 bytes per rappresentare il messaggio MIDI: un intero, Message Type, che rappresenta il tipo di messaggio e il relativo canale MIDI (ad es. 144 = messaggio di tipo NOTE_ON sul canale MIDI 1), e due interi denominati rispettivamente value 1 e value 2, il cui significato varia a seconda del tipo di messaggio (e.g., nel caso di NOTE_ON value 1 rappresenta l'intero relativo al
10 tono della nota suonata, value 2 nel caso di NOTE_ON rappresenta l'intensità con cui la nota viene suonata).

- La SEZIONE C come illustrato in Figura 3 è opzionalmente valorizzato e contiene l'insieme dei valori delle variabili che rappresentano lo stato dello strumento musicale 1 al tempo T presenti nel sender. Secondo una variante
15 realizzativa, la sezione è valorizzata solo nei pacchetti detti di "refresh", altrimenti rimane vuota. Come precedentemente discusso, la lista delle variabili si riferisce preferibilmente alla lista delle variabili che rappresentano note e/o comandi di controllo attivi lato sender al tempo T i.e. che causano la generazione di un suono all'istante T per ogni sender e sono quindi rappresentativi
20 dell'azione del musicista sullo strumento musicale 1 al tempo T. Come mostrato in Figura 4 è possibile rappresentare le informazioni delle variabili con la medesima tripletta che caratterizza i messaggi MIDI: Message Type, value1 e value2. Per nota ancora attiva si intende ad esempio una nota la cui riproduzione ha avuto inizio ad un certo istante e.g. appartenente a Δt_2 in seguito ad un
25 messaggio di tipo NOTE_ON e che è ancora in fase di riproduzione e.g. tramite

l'altoparlante poiché non è ancora stato interrotto da un successivo messaggio di tipo NOTE_OFF. Questo tipo di rappresentazione permette al receiver di confrontare i valori delle proprie variabili di stato relativi a note e/o comandi di controllo al tempo T rispetto alla lista dei valori delle variabili presenti nel pacchetto di refresh ricevuto da ogni sender. Secondo un esempio realizzativo, il receiver aggiorna lo stato delle note e/o degli effetti sonori da riprodurre tramite l'altoparlante sulla base dei valori di ciascuna variabile contenuti nel pacchetto di refresh proveniente dal sender. Ad esempio, valori delle variabili presenti nel pacchetto di refresh sono convertiti in note e/o comandi di controllo ed eseguiti oppure la lista dei valori delle variabili ricevuta dal sender è confrontata con la lista dei valori delle variabili presenti sul receiver al tempo T e questi ultimi sono aggiornati sulla base dei valori dei parametri presenti nel pacchetto di refresh.

Architettura del sistema

Viene descritta una forma preferita di realizzazione per spiegare le interazioni possibili tra i musicisti attraverso un esempio secondo una possibile architettura mostrata in Figura 2. Consideriamo il calcolatore PC1, che svolge il ruolo di MIDI extender ed identificato nel seguito dalla lettera S, come sender, che invia un segnale musicale, e il calcolatore PC2 che riceve il segnale musicale (identificato con la lettera R, come receiver).

Il receiver R:

- i) riceve i dati MIDI dal sender S secondo il formato descritto in precedenza;

- ii) attua, se necessario, il meccanismo di correzione degli errori, e.g. aggiorna i valori delle proprie variabili sulla base di quelli ricevuti dai sender; e
- iii) effettua la riproduzione dei messaggi rappresentativi delle note e/o degli effetti sonori ricevuti.

Nel caso di una comunicazione tra N musicisti, ciascuno si comporterà sia da sender che da receiver secondo quanto descritto in precedenza. In particolare, ogni receiver riceverà dati MIDI e gestirà indipendentemente la comunicazione con ognuno degli N-1 sender remoti e ogni sender invierà dati MIDI in maniera indipendente a ciascuno dei N-1 receiver remoti (adottando quindi un approccio peer-to-peer puro). Per semplicità in Figura 2 è evidenziato il solo traffico relativo al musicista 1.

La descrizione di una possibile realizzazione dell'invenzione fa riferimento alla comunicazione tra un sender ed un receiver. Il sender S è direttamente collegato ad un'interfaccia MIDI disposta ad esempio a bordo dello strumento suonato dal musicista. L'interfaccia MIDI invia dati MIDI asincroni al sender S ogni volta che un messaggio viene generato dall'utente che sta suonando lo strumento. L'intervallo temporale Δt_1 rappresenta la periodicità con cui il sender invia dati e.g. pacchetti ad uno o più receiver. Il sender raggruppa quindi i messaggi ricevuti all'interno di ogni intervallo Δt_1 e, al termine dello stesso, invia al receiver un pacchetto contenente i messaggi che codificano una serie di dati MIDI ricevuti dall'interfaccia MIDI generati nell'intervallo in esame. Nonostante la comunicazione avvenga ogni Δt_1 , che può essere variabile o costante durante la NMP, grazie al campo offset è possibile garantire che la riproduzione dei

singoli messaggi contenuti nel pacchetto avvenga con una granularità temporale più fine rispetto a Δt_1 , (idealmente la stessa usata per la generazione dei messaggi, nel limite della precisione degli orologi e della dimensione del campo offset). Nella scelta del valore da utilizzare per Δt_1 , è importante considerare che

5 un intervallo temporale ridotto abbassa il tempo necessario per la creazione del pacchetto stesso (e dunque concorre a diminuire la latenza complessiva introdotta dal sistema) ma implica l'invio di numerosi pacchetti al receiver con il conseguente aumento del traffico e della quantità dei dati trasmessi. Viceversa, un intervallo temporale più ampio riduce il traffico generato ma comporta un

10 aumento del ritardo percepito dall'utente. Per una applicazione di NMP, un valore ragionevole per Δt_1 è compreso fra circa 5 e 10 ms, preferibilmente non superiore a 15ms. Per quanto riguarda invece la scelta della granularità relativa all'offset dei singoli messaggi inviati in un pacchetto, è necessario che sia inferiore al Δt_1 scelto, ma non in modo eccessivo, dato che oscillazioni dell'istante

15 di riproduzione di un messaggio inferiori ad 1 ms sono impercettibili all'orecchio umano. Una possibile scelta, che concilia le diverse esigenze tra traffico generato e ritardo percepito dall'utente, può essere rappresentata da un offset pari ad 1ms per un $\Delta t_1=10ms$, in generale l'offset non supera il 15% del valore dell'intervallo di tempo. L'invio dei pacchetti è dunque periodico, anche nel caso in cui per

20 talune finestre temporali non viene generato nessun messaggio lasciando in tal caso la sezione B del pacchetto vuota. Questo tipo di approccio permette al receiver di riuscire a distinguere la situazione in cui si verifica la mancanza di ricezione di un pacchetto dovuta ad un ritardo di comunicazione. Al contrario, nel caso in cui il sender invia un pacchetto solo in presenza di messaggi MIDI, il

25 receiver non è in grado di riconoscere la differenza tra l'assenza di messaggi

MIDI ed il ritardo di trasmissione finchè non riceve il pacchetto successivo.

Identificazione e correzione degli errori di trasmissione

Per garantire che la qualità dell'esperienza percepita dall'utente sia tale da soddisfare le caratteristiche richieste nelle applicazioni di NMP, è necessario gestire i casi in cui i pacchetti si perdono oppure siano ricevuti con un ritardo eccessivo e.g. a causa del sovraccarico della rete. Nel caso in cui un pacchetto è ricevuto in ritardo e fuori sequenza, si viene a determinare una condizione di buffer under-run, ovvero quando avviene la riproduzione dell'ultimo pacchetto ricevuto precedentemente senza che ne sono arrivati di nuovi. In questo caso il buffer del receiver resta vuoto a causa dell'arrivo ritardato del pacchetto successivo e l'applicazione non sarà in grado di riprodurre a tempo debito i messaggi in esso contenuti.

Come descritto in precedenza, le informazioni MIDI sono di tipo differenziale, pertanto è di fondamentale importanza per il receiver riprodurle nello stesso ordine con cui sono state generate dal sender, garantendo quindi che i pacchetti siano ordinati secondo il loro numero di sequenza prima che i messaggi in essi contenuti siano riprodotti.

Per permettere al receiver di ricostruire le informazioni perse, ogni sender tiene traccia e memorizza lo stato delle proprie note e/o comandi di controllo, ovvero l'insieme dei valori assunti dalle variabili associate ad ogni nota e/o comando di controllo attivi al tempo T , che con periodicità Δt_2 , include all'interno del relativo pacchetto di refresh.

Analogamente, anche il receiver tiene traccia dello stato dell'insieme delle note

e/o comandi di ogni sender grazie ai pacchetti di refresh che ogni sender invia con periodicità Δt_2 al receiver. In assenza di situazioni di buffer under-run, il receiver ha una rappresentazione dei valori delle variabili dei sender coincidente con i valori delle variabili di ogni singolo sender. Nel momento in cui si verifica
5 una condizione di buffer under-run per uno o più pacchetti contenenti almeno un messaggio, il receiver ha una rappresentazione della lista dei valori delle variabili dei sender disallineata rispetto alla lista dei valori delle variabili del sender da cui ha ricevuto il pacchetto dati. In questo caso il receiver utilizza il pacchetto di refresh per:

10 1) terminare le note e/o i comandi in esecuzione le cui rappresentazioni non sono contenute nella lista delle variabili del pacchetto dati ricevuto dal sender che descrive lo stato dello strumento musicale al tempo T che il sender ha inviato;

2) eseguire le note e/o i comandi la cui rappresentazione è contenuta nella lista delle variabili del pacchetto di refresh ricevuto e che non risultano essere allineati
15 con la rappresentazione del valore delle proprie variabili relative al sender da cui ha ricevuto il pacchetto dati. Nel caso di ricezione del pacchetto, l'errore residuo è legato alla differenza temporale nell'esecuzione delle note e/o comandi rispetto alla situazione che si sarebbe creata in assenza di buffer under-run.

Pertanto per la gestione dei pacchetti dati che arrivano al receiver in ritardo
20 e/o fuori sequenza, per ogni nuovo pacchetto il receiver esegue una routine di controllo come mostrato in Figura 6, dove sono presenti due tipi di controllo:

1) Il controllo del numero di sequenza o generica marca temporale per ordinare i messaggi del pacchetto: se è inferiore a quello di qualsiasi altro

pacchetto ricevuto in precedenza ed il cui contenuto è già stato riprodotto, allora il pacchetto è scartato.

2) Il controllo della presenza della lista di variabili: se non è vuota (ovvero il receiver ha ricevuto un pacchetto di tipo refresh) si procede al confronto dei
5 valori delle variabili che rappresentano le note e/o i comandi dei sender contenuto nella sezione C del pacchetto di refresh inviato dal sender con i valori delle variabili relative ad ogni sender di cui il receiver tiene traccia e attua le correzioni necessarie nel caso in cui i valori non coincidano.

E' infine importante osservare come la scelta della frequenza di invio del
10 pacchetto di refresh Δt_2 è importante ai fini della qualità dell'esecuzione, come mostrato in Figura 7. Scegliere $\Delta t_2 = \Delta t_1$ significa permettere al receiver di correggere al meglio gli errori, ma richiede un aumento significativo della quantità di dati trasmessa.

Secondo una forma preferita di realizzazione, un utente può selezionare un
15 valore preferito di Δt_2 sulla base di un obiettivo di accuratezza della NMP, definito come rapporto fra numero di pacchetti correttamente ricevuti, i.e. non persi o non ricevuti in ritardo e quindi scartati, rispetto al numero di pacchetti complessivi trasmessi. In particolare, è possibile mettere in relazione, ad esempio
tramite una mappa, una probabilità che un pacchetto sia scartato e l'accuratezza
20 della riproduzione audio tramite la rete di scambio dati al variare del rapporto di refresh definito come rapporto fra Δt_2 e Δt_1 . Un utente, sulla base di valori e.g. storici oppure misurati prima della dell'inizio della NMP della probabilità che un pacchetto si scartato a causa del sovraccarico della rete di scambio dati, e dell'accuratezza desiderata, può ottenere tramite tabelle o funzioni interpolanti

il valore di refresh rate adatto.