# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 1 : Bruce Edward Mitchell Inventions, Patents and Patent Applications - Justia Patents Search

*A method of preparing packets for injection into a packet network at an ingress interface of the packet network, for transmission over the network. The method comprises, at the ingress interface, receiving at least two parallel, constant bit rate streams of data, and separately packetising the constant bit rate streams to generate respective packet flows for forwarding to a packet sender.*

Dual-PHY based integrated access device Patent number: A dual PHY-based integrated access device IAD platform employs a highly integrated time division multiplexed TDM , a synchronous transfer mode ATM cell based architecture, to provide enhanced interfacing flexibility for multiple and diverse signaling protocols, effectively reducing the cost and constraints as to choice of host processor used in conventional digital signal processor DSP -based IADs. Grant Date of Patent: April 29, Assignee: Paul Graves McElroy, Phillip Stone Herron, Bruce Edward Mitchell, Darrin Leroy Gieger Echo canceller and compression operators cascaded in time division multiplex voice communication path of integrated access device for decreasing latency and processor overhead Patent number: A cascaded signal processing arrangement processes digitally encoded voice samples transported over a time division multiplex TDM communication path for application to a processor-controlled digital communication device, in particular an integrated access device. Processed voice samples are packetized in accordance with an encapsulating protocol and transmitted as a packetized voice output stream to a destination receiver device. The arrangement includes an echo canceller coupled to the TDM communication path and performing echo cancellation processing on the digitally encoded voice samples. The echo cancellation-processed voice sample signals are compressed by a data compression operator and applied to a packet encapsulating host processor via the TDM communication path. This obviates the necessity of the host processor having to use data bus cycles to download processed digitized voice samples. November 28, Assignee: A voice path direct memory access DMA -based packet generation mechanism writes digitally encoded voice samples directly into prescribed subportions of a preallocated portion of random access memory, to avoid interrupting a main processor for the purpose. A pointer to a respective buffer space subportion is presented to a protocol stack, so that one or more overhead bytes for the stored voice samples can be generated and written into adjacent address space of the preallocated portion of random access memory. The contents of the preallocated memory space are then serialized out for transmission to a destination receiver. June 13, Assignee: Phillip Stone Herron, Bruce Edward Mitchell Method and apparatus for providing reliable voice and voice-band data transmission over asynchronous transfer mode ATM network Patent number: January 24, Assignee: Dual-phy based integrated access device Publication number: December 25, Applicant: September 18, Applicant: Phillip Stone Herron, Bruce Edward Mitchell Echo canceller and compression operators cascaded in time division multiplex voice communication path of integrated access device for decreasing latency and processor overhead Publication number: May 1, Applicant:

# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 2 : Timing > Offloading Traditional Mobile Networks with IP

*The bursty nature of the ATM network and other packet networks has introduced problems in using an adaptive clock recovery scheme to synchronize the local clock at the destination node with the service clock at the source node.*

A multimode clock recovery circuit for use in the provision of constant bit rate services in a cell relay network, comprising: A multimode clock recovery circuit as claimed in claim 1, wherein said receive buffer develops said phase signal in the form of a phaseword based on a fill level thereof. A multimode clock recovery circuit as claimed in claim 1, wherein the phase locked loop goes into a holdover mode when any of the following conditions occur: A multimode clock recovery circuit as claimed in claim 1, wherein said local SRTS generator comprises a divider for receiving a feedback signal from the output of the phase locked loop, a counter for receiving a network clock signal, and a register for generating local SRTS time stamps. A multimode clock recovery circuit as claimed in claim 1, wherein the phase locked loop goes into a holdover mold wherein the output of the phase locked loop remains constant when a valid input signal is lost as to maintain a constant frequency based on the last valid input signal. A phase detector as claimed in claim 16, having an error input for receiving an error flag to generate the phase output signal. A phase detector as claimed in claim 16, further comprising a register connected to the output of said accumulator for temporarily storing the phase output signal. Field of the Invention This invention relates to packet switched networks, and more particularly to a method of clock recovery in cell-relay networks, particularly ATM Asynchronous Transfer Mode networks offering constant bit rate services. CBR services carry time sensitive data, such as voice and video. The original clocked isochronous data is segmented into ATM cells at the transmitting end, carried over the ATM network, and reassembled into a continuous clocked stream at the receiving end. The receiving end needs to provide a clock for this output data stream, or recover the transmitting end clock. Various standardized and un-standardized clocking schemes are current used in ATM: A clock is provided at the receiving end by an incoming data link with the same nominal clock rate as the transmitting end. This incoming clock will often need to be de-jittered before being used, because requirements on outgoing telecom clocks are usually more stringent than requirements on incoming telecom clocks. A telecom clock is recovered from an 8 kHz reference clock. The 8 kHz must be frequency-multiplied up to the desired rate. The clock from is recovered from information about the fill level of the receive buffer containing the received segmented data. This buffer serves to even-out the variation of the incoming data versus the receiver-end clock which is reading out the data. The clock is recovered from time stamps carried by the ATM cells. An object of the present invention is to provide a way of efficiently dealing with all four of these required clocking schemes in one circuit implementation. SUMMARY OF THE INVENTION According to the present invention there is provided a multimode clock recovery circuit for providing constant bit rate services in a cell relay network, comprising an embedded digital phase locked loop PLL including an input circuit capable of generating a phase signal from at least two types of input signal, said phase signal controlling the output of said phase locked loop to generate clock signals for said constant bit rate services. The same PLL is used for all three kinds of inputs thereby providing a gate efficient implementation. The present invention is primarily concerned with telecom clocks at T1, E1 and J2 rates, as well as TDM bus clocks at 2. The invention is also applicable to DS3 and E3 clock rates. The clock recovery circuit in accordance with the invention efficiently supports all four clock schemes. The SRTS scheme recovers phase of original clock, not just the frequency. This will vary the inter-arrival times of the cells, as the cells contain either 46 or 47 bytes, but the number of DSO channels carried by the 46 or 47 bytes vary. This invention can correctly recover timing from the Receive Buffer even as the cell inter-arrival times change. The invention can also offer an accurate holdover mode in the STRS mode, which is controllable by information derived from received ATM cells. For example, an SRTS time stamp may be detected to be invalid because it is protected with error-detection fields. The clock recovery circuit of the invention will automatically go into accurate holdover mode when the all-idle

state is detected as defined by the DBCES channel activity bit mask. The invention can also provide an accurate freerun clock 0. The ability to do Line Rate and SRTS Clocking in one PLL means that at the transmitting end the present invention can be used to de-jitter the incoming clock before it is used to generate SRTS time stamps, and at the receiver another instance of the present invention can be used to generate the outgoing clock from the SRTS. This is important, because the sampling effects of the time stamp process will alias the jitter frequencies downward, where they will be harder to filter for the receiver PLL creating wander. The incoming ATM cell stream is reassembled in reassembly engine 3. In the adaptive scheme shown in FIG. This is important because the sampling effects of the time stamp process alias the jitter frequencies downward, where they are harder to filter for the receiver PLL, which creates wander. As noted above, the PLL 1 operates in the following modes: The PLL synchronizes to the incoming clock. Synchronize to the incoming cell stream by monitoring the write and read pointers of the Reassembly Circular Buffer. The PLL receives a phaseword representing: This mode is similar to Synchronous mode. An 8 kHz clock, generated from the  The output can be a line rate cock or an SRTS. Within the modes listed above, the PLL 1 has another set of basic modes: The PLL synchronizes using the appropriate input signal i. The PLL no longer uses the input signal, but holds to its current output frequency. The user can also switch the PLL into holdover mode. No input is used. The master clock accuracy determines the output clock accuracy. The digital PLL 1 must provide a clock synchronous to some input signal. Because there are several types of incoming signals to synchronize to such as clocks, SRTS data and arriving ATM cells , the PLL needs to be capable of accepting various types of input. The basic PLL architecture, however, is the same for all situations. As shown in FIG. Parameters such as locking range, center frequency, transfer, accuracy, etc. In addition, features like freerun and holdover mode can be built in. When the system clock is high enough, no analog circuitry is needed to meet the output jitter requirements. A divider on the  The PLL must generate frequencies with an accuracy as stated in Table  The accuracy defines the freerun accuracy and the locking range of the PLL. The accuracy of mclk must be incorporated in these calculations. A freerun accuracy of approximately 0. For DS1, the locking range is approximately ppm, thereby supporting older systems with a ppm frequency accuracy.

# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 3 : Adaptive clock recovery and jitter control in ATM networks - UBC Library Open Collections

*EAL-TIME variable bit rate (VBR) services carried over asynchronous packet networks impose serious challenges on the source clock frequency recovery (SCFR) at a receiver.*

What is claimed is: A method for clock recovery in a packet network, the method comprising: The method of claim 1, wherein receiving data packets at the destination node comprises receiving data packets at a reassembler, wherein the reassembler places the data packets in proper sequence then transmits the data packets to the buffer. The method of claim 2, wherein storing the data packets in the buffer comprises, writing data from the data packets into the buffer. The method of claim 3, and further comprising passing the data from the buffer to a Line Interface Unit, without altering payload of the data packets. The method of claim 4, wherein monitoring a fill level of the buffer comprises monitoring a difference between a read address and a write address of the buffer. The method of claim 5, wherein identifying a relative maximum fill level flyer comprises storing a value in a register. A node in a telecommunications network, comprising: The node of claim 8, wherein the reassembler separates data packets and puts the data packets in proper sequence. The node of claim 9, wherein the peak fill level detector includes a register that is continuously updated over a period of time with a relative maximum buffer fill level. The node of claim 8, wherein the variable oscillator comprises a numerically controlled oscillator. The node of claim 8, wherein the variable oscillator comprises a direct digital synthesis oscillator. A method for adaptive clock recovery, the method comprising: The method of claim 14, and further comprising adjusting the value for the numerically controlled oscillator to reduce phase uncertainty. The method of claim 14, wherein monitoring a buffer fill level comprises comparing read and write addresses for the buffer. A destination node implementing adaptive clock recovery, the node comprising: The node of claim 17, wherein the peak fill level detector is implemented in a Field Programmable Gate Array. The node of claim 17, wherein the oscillator comprises a numerically controlled oscillator. The node of claim 17, wherein the processor further implements phase tracking for the oscillator. By using ATM, various services including voice, video, and data, can be multiplexed, switched, and transported together in a universal format thus permitting network resources to be shared among multiple users. The full integration of various services may also allow simpler and more efficient network and service administration and management. Accommodation of CBR services is, however, an important feature of ATM, both for universal integration and for compatibility between existing and future networks. Typically, a source node sends data regulated by a service clock through an ATM network to a destination node. A clock controlling a destination node buffer must operate at a frequency matched to that of the service signal input at the source node to avoid loss of information. Thus, when a CBR service is implemented in a packet network, such as an ATM network, a buffer is used at the destination node to store data temporarily. The data in the buffer is read out at a constant bit rate established by a local clock at the destination node. The bursty nature of the ATM network and other packet networks has introduced problems in using an adaptive clock recovery scheme to synchronize the local clock at the destination node with the service clock at the source node. ATM networks introduce random delays in the transmission of data packets between two nodes. Unfortunately, the CTDV may introduce significant wander components into a clock signal at a destination node that uses an adaptive clock recovery scheme. When significant wander components are contained in the CTDV, the clock signal generated by current adaptive clock recovery schemes at the destination node may follow CTDV, not the service clock of the source node. A circuit and method for adaptive clock recovery for CES that uses a peak buffer fill level as an indicator to lock a local clock at a destination node with the service clock at a source node is disclosed. In particular, an illustrative embodiment of the present invention includes a method for clock recovery in a packet network. The method includes a network which receives data packets at a destination node. The data packets are stored in a buffer. The data packets are read out of the buffer by using a locally generated clock. The fill level of the buffer is monitored over a first period of time. A relative

maximum fill level for the buffer is identified during the first period of time. Further, the relative maximum fill level is used to control the frequency of the locally generated clock so as to control the rate at which data is read out of the buffer. This unique clock control algorithm and mechanism produces a recovered clock which contains no jitter. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical and electrical changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense. A source node sends data packets through an ATM network

# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 4 : USB2 - Adaptive clock recovery - Google Patents

*Hence, in many cases the only alternative is to attempt to recover the clock based exclusively on the TDMoIP traffic, a technology known as "adaptive clock recovery". This is possible since the source TDM device is producing bits at a constant rate determined by its clock, although this rate is hidden by the PDV.*

Background[ edit ] Communications service providers and enterprise customers are interested in deployment of voice and leased line services over efficient Ethernet, IP and MPLS infrastructures. TDMoIP presents a migration path, whereby modern packet switched networks can be used for transport, while the end-user equipment need not be immediately replaced. In , the IETF set up the PWE3 working group , which was chartered to develop an architecture for edge-to-edge pseudowires, and to produce specifications for various services, including TDM. TDM framing locks the frame rate to the sampling frequency of voice traffic, so that there are always frames per second; a T1 frame consists of bits and an E1 frame of bits. Unlike unframed TDM for which all bits are available for payload, framed TDM requires dedicating of some number of bits per frame for synchronization and perhaps various other functions e. Framed TDM is often used to multiplex multiple voice channels each consisting of 8-bit samples per second in a sequence of timeslots recurring in each frame. When this is done we have "channelized TDM" and additional structure must be introduced. In order to efficiently transport slowly varying channel associated signalling bits, second order structures known as multiframes or superframes are defined. For example, for E1 trunks the CAS signaling bits are updated once per multiframe of 16 frames every 2 milliseconds while for T1 ESF trunks the superframe is 24 frames 3 milliseconds. Other types of second order structures are also in common use. TDMoIP transport is denoted "structured-agnostic" when the TDM is unframed, or when it is framed or even channelized, but the framing and channelization structure are completely disregarded by the transport mechanisms. In such cases all structural overhead must be transparently transported along with the payload data, and the encapsulation method employed provides no mechanisms for its location or utilization. Structure-aware TDM transport may explicitly safeguard TDM structure, in three conceptually distinct ways, which we shall call structure-locking, structure-indication, and structure-reassembly. Structure-indication allows packets to contain arbitrary fragments of basic structures, but employs pointers to indicate where the following structure commences. In structure-reassembly components of the TDM structures may be extracted and reorganized at ingress, and the original structure reassembled from the received constituents at egress. Adaptation denotes mechanisms that modify the payload to enable its proper restoration at the PSN egress. By using proper adaptation, the TDM signaling and timing can be recovered, and a certain amount of packet loss can be accommodated. Encapsulation signifies placing the adapted payload into packets of the format required by the underlying PSN technology. These are the standard headers used by the PSN technology, e. The control word contains a bit packet sequence number needed to detect packet re-ordering and packet loss , payload length, and flags indicating defect conditions. A side benefit of this choice of payload types is simplified interworking with circuit emulation services carried over ATM networks. AAL1 operates by segmenting the continuous stream of TDM data into small byte cells and inserting sequencing, timing, error recovery, and synchronization information into them. By allowing multiple cells per packet, TDMoIP facilitates flexible tradeoffs of buffering delay which decreases with fewer cells per packet for bandwidth efficiency which increases with more cells per packet, due to the per packet overhead. AAL2 operates by buffering each TDM time slot into short minicells, inserting the time slot identifier and length indication, sequencing, and then sending this minicell only if it carries valid information. TDMoIP concatenates the minicells from all active time slots into a single packet. Delay[ edit ] The telephony network severely constrains end-to-end delays. These constraints are not problematic for TDM networks, where the major component of the end-to-end delay is electrical propagation time "light speed delay". By contrast, IP-based systems typically add various forms of delay, one of which is based on the time it takes to form packets packetization delay , which is proportional to the packet

size divided by the data rate. Packet sizes cannot be made too small or the packet header overhead will become overwhelming. The other form of delay introduced by IP systems is the playout delay, which needs to be added at the recipient to buffer packet delay variation and ensure a smooth playout. VoIP systems that try to be very bandwidth efficient may also add tens of milliseconds of algorithmic delay in the voice codec. Historically, bad implementations have added additional, operating-system induced delays, which together with the other delays in practice sometimes approach ms even before taking propagation delays into account. The packetization latency added by TDMoIP depends on the number of cells per packet but is typically in the single millisecond range due to the higher data rate of a complete multiplex as compared to a single VoIP flow. Playout delay considerations do not differ materially between TDMoIP and VoIP, however, so both work best on paths with controlled packet delay variation strong overprovisioning or "QoS". Timing recovery[ edit ] Native TDM networks rely on hierarchical distribution of timing. This node, which offers Stratum 1 accuracy, provides the reference clock to secondary nodes with Stratum 2 accuracy. The secondary nodes then provide a time reference to Stratum 3 nodes. This hierarchy of time synchronization is essential for the proper functioning of the network as a whole. Packets in the PSN reach their destination with delay that has a random component, known as packet delay variation PDV. When emulating TDM transport on such a network, this randomness may be overcome by placing the TDM packets into a jitter buffer from which data can be read out at a constant rate for delivery to TDM end-user equipment. The problem is that the TDM source time reference is no longer available, and the precise rate at which the data are to be "clocked out" of the jitter buffer is unknown. Since each of these clocks is highly accurate, they necessarily agree to high order. Hence, in many cases the only alternative is to attempt to recover the clock based exclusively on the TDMoIP traffic, a technology known as "adaptive clock recovery". This is possible since the source TDM device is producing bits at a constant rate determined by its clock, although this rate is hidden by the PDV. The task of clock recovery is thus an "averaging" process that negates the effect of the random PDV and captures the average rate of transmission of the original bit stream. Packet loss[ edit ] While proper application of traffic engineering and quality-of-service QoS is expected to minimize packet loss, packets will at times arrive at the egress out of order. They may also have been dropped altogether within the PSN. The TDMoIP control word described above includes a bit sequence number for detecting and handling lost and mis-ordered packets. Misordered packets may be either reordered or dropped and interpolated. While the insertion of arbitrary packets may be sufficient to maintain the TDM timing, in voice applications packet loss can cause gaps or errors that result in choppy, annoying, or even unintelligible speech. The precise effect of packet loss on voice quality and the development of packet loss concealment algorithms have been the subject of detailed study in the VoIP community, but their results are not directly applicable to the TDMoIP case. This is because VoIP packets typically contain between 80 samples 10 ms and samples 30 ms of the speech signal, while TDMoIP packets may contain only a small number of samples. Since TDMoIP packets are so small, it is acceptable to simply insert a constant value in place of any lost speech samples. Assuming that the input signal is zero-mean i. Alternatively, more sophisticated approaches call for optimally predicting the values of missing samples.

## Chapter 5 : MÃ¡rio Jorge LeitÃ£o | Universidade do Porto - blog.quintoapp.com

*A method of recovering a clock signal for a TDM output from packets of TDM data which have been transmitted over a packet network, from a source having a source TDM clock to a destination having a destination TDM clock.*

These architectures provide different levels of resiliency, complexity, and cost. In general, the following principles are adhered to in all PRC architectures: The synchronization distribution is tree-shaped. The synchronization network can be decomposed into multiple synchronization chains. A higher-quality level is never slaved to a reference signal of a lower-quality. The SSU provides timing to a portion of the network. PRCs use a combination of local Cesium tubes and radio-based Cesium tubes. SSUs receive clocking information from higher-layer clocks and distribute the clock information to all local equipment. SSUs also have the ability to provide accurate holdover mode, in the event that their clock source is lost. The SSU does not belong to the transport network, but only provides the timing for the transport network elements within its synchronization domain. There are two primary methods for providing clock synchronization, as follows: Master-slave synchronization, which has a single PRC from which all other clocks are synchronized. Synchronization in this method is achieved by sending timing signals from one clock to the next, in a hierarchical fashion. Figure illustrates this master-slave synchronization network architecture. Figure Master-Slave Synchronization Network Mutual synchronization, in which all clocks are interconnected. In this method, there is no unique PRC or hierarchical structure defined. Figure illustrates this mutual synchronization network architecture. Figure Mutual Synchronization Network In practice, master-slave and mutual synchronization methods may be deployed in combination. Synchronization from the main PRC is done in standard master-slave hierarchical fashion. Figure illustrates this combined network architecture. Priorities for clock source and synchronization are identified in the figure. Network elements may operate in four different timing modes, as follows: External timing, where the reference source signal is received via a local timing interface directly. Line timing, where the reference source signal is received from one or more data interfaces that also carries timing information. Loop timing, where the reference source signal is received from only one data interface as part of a ring topology. Through timing, where the reference source signal is transported transparently across the network element. These timing modes map to four network architecturesâ€”synchronous networks, asynchronous networks, pseudo-synchronous networks, and plesiochronous networks. Synchronous A synchronous network is one where all clocks within the network have identical long-term accuracy. These networks require synchronization to avoid jitter and wander. Synchronous networks have a single active PRC source signal and rely on line timing to distribute clock information across the network. Figure depicts a synchronous network that relies on line timing for clock source. Figure Synchronous Network Asynchronous An asynchronous network is one where not all clocks within the network have identical long-term accuracy due to multiple clock sources. In an asynchronous network, clocks are operating in free-running mode. These networks do not require that all clocks be synchronized to operate properly. Figure depicts an asynchronous network. These networks require synchronization to work properly. Figure depicts a pseudo-synchronous network. Figure Pseudo-Synchronous Network Plesiochronous A plesiochronous network is one where different parts of the network are not perfectly synchronized with each other. Plesiochronous networks operate within a threshold of acceptable asynchronization; that is, two network elements act as if they are synchronized, but must accept and cope with time slips. A plesiochronous network is depicted in Figure  All network components must be synchronized with each other to ensure that data is not lost. In a native TDM network, clock synchronization is performed at the physical layer, and clocking information is carried along with data traffic. Clock slips occur when the receiver and transmitter have clocks that either run faster or slower than the other. These clock slips result in frames being either added or lost from the data stream. IP networks, by nature, are asynchronous, and therefore cannot provide a constant bitrate. Due to this, it is not possible to use the physical layer clock

synchronization information from the native TDM frame for accurate clocking over pseudowires. Although pseudowire endpoints do not need the clock synchronization information directly to implement the packet-switching functions, the constant bitrate applications that leverage the pseudowire transport must receive accurate timing information. This requires that the packet-switched networkâ€"that is, the pseudowire itselfâ€"provide this information to the applications. In such architecture, the reference clock may be connected directly to the synchronous network elements on each side of the pseudowire see Figure or to the pseudowire interworking function see Figure  Clock Recovery over Packet Clock recovery is an important consideration when providing circuit emulation services over a PSN. There are two methods to provide clock recovery over packet, as follows: Differential Clock Recovery involves having a reference clock available at both sides of the pseudowire. Only the difference between the reference clock and the IWF service clock is transmitted across the pseudowire. Although this solution provides accurate frequency information and is tolerant to network delay, delay variation jitter , and packet loss, the differential clock recovery solutions are expensive because they require multiple reference clocks. Figure illustrates differential clock recovery. Figure Differential Clock Recovery Adaptive Clock Recovery involves having a reference clock available only at one side of the pseudowire. A timestamp is applied to all outbound packets by the sending IWF. The receiving IWF uses the information in the timestamp to recover the original reference clock information. Although this solution is less expensive only a single reference clock is required , adaptive clock recovery is more susceptible to delay variation. TDMoIP uses adaptive clock recovery. Figure illustrates adaptive clock recovery. Figure Adaptive Clock Recovery Timing over Packet Solutions There are four technologies for addressing synchronization over a packet network: Any timing protocol should operate over the generic Internet with little or no intervention or management. Due to the unpredictable nature of the Internet, however, the accuracy of the protocol is greatly diminished. The accuracy of the frequency and time distribution is improved when operated over a managed network. SyncE Synchronous Ethernet SyncE is a line-timing method for transporting timing information over the Ethernet physical layer. SyncE specifications and requirements rely on four primary standards, as follows: Synchronization layer functions SyncE standards provide additional functionality to the  Downstream clocks use the SSM for troubleshooting purposes, as the SSM will communicate if the clock source is a synchronized signal or derived from a free-running oscillator. Four types of clocks are defined within PTP, as follows: The OC may be a master or slave, and may be responsible for providing time to an end node or application. These interfaces may consist of multiple master interfaces, but only a single slave interface. The BC transfers all timing on the slave interface to the master interfaces. The BC can only be responsible for providing time to an application, not an end node. There are two types of transparent clocks: Prior to synchronization, the clocks are organized into a master-slave hierarchy through a series of PTP Announce messages. The hierarchy contains a grandmaster, or PRC, multiple masters, and multiple slaves. This selection process is the Best Master Clock Algorithm BMCA , which includes a clock class, based on where the clock has synchronized its timing from; clock accuracy, based on maximum accuracy threshold; and time source, based on the type of clock from which the advertising clock has received its timing Atomic, GPS, Terrestrial Radio, PTP, Internal oscillator, and so on. Synchronization in PTP Once the hierarchy is established, each slave then synchronizes with its master using either a Delay Request-Response mechanism or a Peer Delay mechanism. These mechanisms cannot be mixed over the same communications path. Figure illustrates this Request-Response mechanism. Figure PTP Delay Request-Response Mechanism Propagation time, or transit time, and an offset, or processing time, are calculated during this process. Assuming a symmetrical link: Assuming an asymmetrical link: The propagation time is the average of the slave-to-master and master-to-slave propagation times. The offset is the difference between the actual master-to-slave time and the average propagation times. The Peer Delay mechanism is also symmetric; that is, it operates separately in both directions. The Peer Delay mechanism consists of five messages: Figure PTP Peer Delay Mechanism A propagation time, or the transit time, and an offset, or the processing time, are calculated during this process. This time is the residence time, and is

included in the message as a Correction field, such that OC and BC may account for this processing time. Each E2E TC in the chain adds its own residence time to the value already contained in the Correction field. PTP Profiles and Conformance PTP supports extensible profiles that allow for transport of optional features and attribute values, including interworking and desired performance levels required for a particular application. These profiles are created by numerous third parties, such as standards or industry organizations and vendors. IEEE defines two default profiles: In addition, a network node may comply with certain optional sections of the standards but must implement the optional section in its entirety. The most recent version, NTPv4, extends upon previous versions NTPv3â€”RFC by introducing accuracy to the tens of microseconds with a precision time source, such as a Cesium oscillator or GPS receiver , dynamic discovery of servers, and includes an extensibility mechanism via options. Primary servers synchronize to national time standards via radio terrestrial or satellite. A client synchronizes to one or more upstream servers, but does not provide any synchronization services to downstream nodes. A Secondary server synchronizes to one or more Primary servers and also provides synchronization services to one or more downstream servers or clients.

# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 6 : TDMoIP - Wikipedia

*One requirement for ATM networks is the support for the provision of any constant and continuous bit rate service. This paper examines the evolution of the different alternatives considered within.*

Intrastream Synchronization for Continuous Media Streams: The transmission of real-time streams over best-effort networks has been an interesting research area for over a decade. An important objective of the research community has been to devise methods that cope with the variations of the network delay -- also called delay jitter -- that are an inherent An important objective of the research community has been to devise methods that cope with the variations of the network delay -- also called delay jitter -- that are an inherent characteristic of best-effort networks. Jitter destroys the temporal relationships between periodically transmitted media units MUs that constitute a real-time media stream, thus hindering the comprehension of the stream. Playout adaptation algorithms undertake the labor of the temporal reconstruction of the stream, which is sometimes referred to as the restoration of its intrastream synchronization quality. This paper surveys the work in the area of playout adaptation, aiming to concisely organize ideas that have been presented in isolation and identify the main points of differentiation amongst different schemes. The survey discusses issues related to the timing information, the handling of late media units, the quality evaluation metrics, and the adaptation to changing delay conditions. The prevalence of the IEEE  It thus becomes rather difficult to configure an appropriate de-jitter buffer to maintain the temporal fidelity of the AV presentation. We propose in this paper an adaptive delay and synchronization control scheme for AV conferencing applications over campus-wide WLANs. Making use of a distributed timing mechanism, the scheme monitors the synchronization errors and estimates the delay jitters among adjacent Media Data Units MDUs in real-time. It piece-wisely controls the equalization delay to compensate for the delay jitters experienced by MDUs in a closed-loop manner. We investigate the performance of the proposed scheme through trace-driven simulations. We collected network traces from a production campus-wide IEEE  Simulation results show that the scheme is capable of dynamically balancing between synchronization requirements and latency requirements in all scenarios. Small synchronization phase distortions, low MDU loss percentages and low average end-to-end delay can be achieved simultaneously. There have only been a few reported attempts aimed at low-latency multi-object applications such as AV conferencing. It lacks generality due to this unrealistic assumption on the Internet Making use of a distributed timing mechanism, the scheme monitors the synchronization errors and estimates the delay jitters among adjacent MDUs in real-time. It piece-wisely controls the equalization delay to compensate for the delay jitters experienced by MDUs in a closedloop manner. In particular, compared with solutions using a static setting, the proposed scheme is able to achieve a gain of around ms in end-to-end delay with the same amount of MDU losses under some mediaunfriendly situations. Show Context Citation Context Networking , " Abstractâ€”When multimedia streams arrive at the receiver, their temporal relationships may be distorted due to jitter. There are various ways to reduce jitter, which include There are various ways to reduce jitter, which include synchronization at the application layer, or synchronization at the asynchronous transfer mode ATM adaptation layer AAL. The new source rate recovery scheme called jitter time-stamp JTS provides synchronization at the ATM adaptation layer 2 AAL2 which is used to carry variable bit-rate traffic such as compressed voice and video. JTS is implemented, and experiments have shown that it is able to recover the source rate. Due to the lack of QoS support, ensuring an acceptable application level QoS for the real-time delivery of multiobject multimedia presentations on the current Internet is very challenging. In this paper, we first investigate the application-level objective performance measures which reflect the user-perceived quality, and then propose an adaptive delay and synchronization control scheme making use of those metrics in real-time. We take a wireline video conferencing application consisting of one audio stream and one video stream as an example. We detail the operation of the proposed scheme and investigate the resultant performance. The simulation results show that small synchronization phase distortions, low MDU

loss percentages and low average end-to-end delays could be achieved simultaneously. We further discuss a more challenging issue raised by mobile wireless applications. A preliminery solution is proposed. Abstract â€" In this paper, we present an analytical study for the Gilbert loss model describing temporally correlated loss observed in the Internet and other communication systems. We obtain closed-form solutions describing transient and steady-state behav-ior of the model. Additionally starting from recursive equations of the Gilbert loss model, we derive closed-form solutions for ar-rival and loss patterns of the systems governed by the model. We show that utilizing our model yields to a lower complexity com-pare to the existing recursive models, attracting special attention of many different wired and wireless networking applications re-lying on such a model. The model has also been in wide spr In this paper, we propose an adaptive synchronization control scheme to achieve both intra-stream and interstream synchronization for real-time streaming multimedia applications on the Internet. Based upon synchronization errors calculated in real-time, our scheme piecewisely adjusts the end-to-end Based upon synchronization errors calculated in real-time, our scheme piecewisely adjusts the end-to-end delay to compensate for the delay variation. It does so by controlling the virtual clock implemented at the destination end. Simulation results show that our control scheme is able to maintain good synchronization performance under different network conditions. In order to enable a truly pervasive computing environment, next generation networks including B3G and 4G will merge the broadband wireless and wireline networking infrastructure. However, due to the tremendous complexity in administration and the unreliability of the wireless channel, provision o However, due to the tremendous complexity in administration and the unreliability of the wireless channel, provision of hard-guarantees for services on such networks will not happen in the foreseeable future. This consequently makes it particularly challenging to offer viable AV conferencing services due to their stringent synchronization, delay and data fidelity requirements. Expecting no special treatment from the network, we apply a novel adaptive delay and synchronization control mechanism to maintain the synchronization and reduce the latency as much as possible. We also employ a robust video coding technique that has better error-resilience capability. We investigate the performance of the proposed solution through simulations using a three-state hidden Markov chain as the generic end-to-end transport channel model. The results show that our scheme yields tight synchronization performance, relatively low end-to-end latency and satisfactory presentation quality. The scheme successfully provides a fairly robust AV conferencing service.

# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 7 : CSDL | IEEE Computer Society

*Adaptive Clock Recovery 29 3 Adaptive Clock Recovery In real-time CBR applications, the convergence sublayer of the A A L - 1 layer at the receiver is responsible for the recovery of the transmitter clock and removing the jitter from the arriving cell stream.*

A simple first order Clock Control Algorithm is given by the following difference equation: It is selected to track long term drift in the remote clock frequency, fservice, but reject short term variation due to packet delay variations. An enhanced Clock Control Algorithm is given by the following equation: This has the effect of controlling phase shift between the remote and local clocks. Alternative Filter algorithms may be used. Alternative Clock Control Algorithms may be used e. The method as described makes use of all the data packets. It is also possible to use a subset of the packets, or to use special timing packets. It would be possible to implement the method without remote timestamps where packets are of a consistent payload size, provided a sequence number is available in the packets transmitted from the source to the destination. The remote timestamp can be reconstructed at the destination by multiplying the sequence number by the size of the TDM payload. As mentioned above, the size may be expressed in fractions of a bit, bits, bytes, frames or packets. The use of a sequence number applied by the source device ensures that the timestamp calculation is not corrupted by lost or out-of-sequence packets. The method has application in timing recovery over packet based systems or other asynchronous systems. Circuit emulation may be used to support the provision of leased line services to customers using legacy TDM equipment. The advantages are that a carrier can upgrade to a packet switched network, whilst still maintaining their existing TDM business. The clock recovery method described above provides the following advantages: The method is able to make use of all of the incoming data packets at the destination device to converge average fregen i. No expensive Clock Generation Circuits are required such as oven controlled crystal oscillators. Timestamps contained in the incoming packets are compared to a Local Timestamp value to obtain a Transit Time value. The sequence of Transit Time values that is obtained are filtered. The separation of the filter from the Clock Recovery Algorithm allows the Clock Control Algorithm to operate at a much slower rate than the filter. So that, for example, a high speed filter could be implemented in Hardware and a low speed Clock Control Algorithm with an external CPU. This confers significant benefits, such as flexibility, reduction of development risk, ease of optimising the solution for a specific environment etc. A method is provided to enable the Local Timestamp to be initialised to the value of the first received Remote Timestamp to minimise wraparound problems and prevent a start up error with the recovered clock. A method is provided to allow packets to be deleted from the PDV Buffer and dummy packets to be inserted into the PDV Buffer in order to adjust the device latency. This does not affect the Local Timestamp value mentioned above. The clock recovery method described here allows the PDV Buffer depth to be varied independently of the clock recovery mechanism. This allows the clock recovery to stabilise prior to setting up the PDV Buffer, and allows the buffer to be changed during operation to match any underlying shift in network characteristics. It is also possible to exclude late packets from the Transit Time calculations, which may improve performance. Such packets may artificially increase the Transit Time, causing the recovered clock to appear as though it is running too fast. A method of recovering a clock signal for a TDM output from packets of TDM data which have been transmitted over a packet network, from a source having a source TDM clock to a destination having a destination TDM clock, the method comprising: A method as claimed in claim 1 , wherein a filter is provided to filter said Transit Time value over time. A method as claimed in claim 2 , wherein said filter is a first order low pass filter. A method as claimed in claim 1 , wherein the received packets are placed in the packet buffer, and the buffer depth is controlled by a depth control algorithm. A method as claimed in claim 4 , wherein said depth control algorithm makes adjustments to said packet buffer by adding or removing packets. A method as claimed in claim 1 , wherein said Remote Timestamp is calculated at said destination by counting the number of packet

payload bits which have been received. A method as claimed in claim 1 , wherein a sequence number is allocated sequentially to each packet, and wherein said Remote Timestamp is calculated at said destination by multiplying the packet payload size by the packet sequence number. A method as claimed in claim 1 , wherein said clock frequency is controlled by a clock control algorithm which ensures that the change in said clock frequency is proportional to the change in the average transit time. A method as claimed in claim 8 , wherein said clock control algorithm is given by: A method as claimed in claim 8 , wherein said clock control algorithm also incorporates phase locking between the source and destination TDM clocks. A method as claimed in claim 10 , wherein said clock control algorithm is given by: A clock recovery system for recovering a clock signal for a TDM output from packets of TDM data which have been transmitted over a packet network, from a source having a source TDM clock to a destination having a destination TDM clock, the system comprising: A clock recovery system as claimed in claim 12 , wherein said filter is a first order low pass filter. A clock recovery system as claimed in claim 12 , which further comprises the packet buffer for holding received packets, and a depth control arrangement for controlling the depth of the packet buffer. A clock recovery system as claimed in claim 14 , wherein said depth control arrangement is arranged to make adjustments to said packet buffer by adding or removing packets. A clock recovery system as claimed in claim 12 , wherein said remote timestamp extraction means calculates said Remote Timestamp by counting the number of packet payload bits which have been received. A clock recovery system as claimed in claim 12 , wherein a sequence number is allocated sequentially to each packet, and wherein said Remote Timestamp is calculated at said destination by multiplying the packet payload size by the packet sequence number.

# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 8 : Adaptive clock recovery for circuit emulation service - ADC Telecommunications, Inc.

*IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 3, NO. 6, DECEMBER An Adaptive Congestion Control Scheme for Real Time Packet Video Transport Hemant Kanakia, Partho P. Mishra, and Amy R. Reibman, Member, IEEE.*

Current uses[ edit ] Post-production houses, content delivery networks and studios use adaptive bit rate technology in order to provide consumers with higher quality video using less manpower and fewer resources. The creation of multiple video outputs, particularly for adaptive bit rate streaming, adds great value to consumers. Media companies have been actively using adaptive bit rate technology for many years now and it has essentially become standard practice for high-end streaming providers; permitting little buffering when streaming high-resolution feeds begins with low-resolution and climbs. Benefits of adaptive bitrate streaming[ edit ] This section does not cite any sources. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. The media and entertainment industry also benefit from adaptive bitrate streaming. As the video space grows, content delivery networks and video providers can provide customers with a superior viewing experience. Adaptive bitrate technology requires additional encoding , but simplifies the overall workflow and creates better results. HTTP-based adaptive bitrate streaming technologies yield additional benefits over traditional server-driven adaptive bitrate streaming. Second, since HTTP streaming is purely client-driven, all adaptation logic resides at the client. This reduces the requirement of persistent connections between server and client application. Furthermore, the server is not required to maintain session state information on each client, increasing scalability. The CDN receives the stream from the source at its Origin server, then replicates it to many or all of its Edge cache servers. The end-user requests the stream and is redirected to the "closest" Edge server. The use of HTTP-based adaptive streaming allows the Edge server to run a simple HTTP server software, whose licence cost is cheap or free, reducing software licensing cost, compared to costly media server licences e. Adobe Flash Media Streaming Server. The original XML schema provided a simple playlist of bit rates, languages and url servers. Adobe HTTP Dynamic Streaming[ edit ] "HTTP Dynamic streaming is the process of efficiently delivering streaming video to users by dynamically switching among different streams of varying quality and size during playback. This provides users with the best possible viewing experience their bandwidth and local computer hardware CPU can support. Another major goal of dynamic streaming is to make this process smooth and seamless to users, so that if up-scaling or down-scaling the quality of the stream is necessary, it is a smooth and nearly unnoticeable switch without disrupting the continuous playback. HTTP-based streaming also allows video fragments to be cached by browsers, proxies, and CDNs , drastically reducing the load on the source server. HLS supports both live and Video on demand content. It works by breaking down streams or video assets into several small MPEG2-TS files video chunks of varying bit rates and set duration using a stream or file segmenter. One such segmenter implementation is provided by Apple. Each playlist pertains to a given bitrate level, and contains the relative or absolute URLs to the chunks with the relevant bitrate. The client is then responsible for requesting the appropriate playlist depending on the available bandwidth. These adaptive streams can be made available in many different bitrates and the client device interacts with the server to obtain the best available bitrate which can reliably be delivered. Microsoft has successfully demonstrated delivery of both live and on-demand p HD video with Smooth Streaming to Silverlight clients. The streaming server is an HTTP server that has multiple versions of each video, encoded at different bitrates and resolutions. The control is entirely server-based, so the client does not need special additional features. The streaming control employs feedback control theory. Rather than streaming and storing multiple formats for different platforms and devices, upLynk stores and streams only one. While most of the initial self-learning approaches are implemented at the server-side [39] [40] [41] e. In all of these approaches, the client state is modeled using, among others, information about the current perceived network throughput

and buffer filling level. Based on this information, the self-learning client autonomously decides which quality level to select for the next video segment. The learning process is steered using feedback information, representing the Quality of Experience QoE e. Furthermore, it was shown that multi-agent Q-learning can be applied to improve QoE fairness among multiple adaptive streaming clients. Some of the documented considerations are things such as additional storage and encoding costs, and challenges with maintaining quality globally. There have also been some interesting dynamics found around the interactions between complex adaptive bit rate logic competing with complex TCP flow control logic. This also proves to be a problem with digital rights management being employed by any streaming protocol. The method of segmenting files into smaller files used by some implementations as used by HTTP Live Streaming could be deemed unnecessary due to the ability of HTTP clients to request byte ranges from a single video asset file that could have multiple video tracks at differing bit rates with the manifest file only indicating track number and bit rate.

# DOWNLOAD PDF AN ADAPTIVE CLOCK RECOVERY SCHEME FOR CONSTANT BIT RATE REAL-TIME SERVICES

## Chapter 9 : Adaptive bitrate streaming - Wikipedia

*Real-time services as well as mobile technologies require (channel clock) to C (bit-rate allocated in assembly) i.e. clock recovery (frequency) For time.*

The method comprises, at the ingress interface, receiving at least two parallel, constant bit rate streams of data, and separately packetising the constant bit rate streams to generate respective packet flows for forwarding to a packet sender. The frequency of at least one of the packet flows with respect to another packet flow is set so as to reduce the degree of correlation between the packet flows, and the packet flows sent to the same or different egress interfaces over the packet network. The invention is applicable in particular, though not necessarily, to the synchronisation of clocks associated with time division multiplexed transmission links interconnected by a packet network. Older systems tend to use the former, and in the main use time division multiplexing to divide the time domain, for a given frequency band, into time slots of equal duration. Circuits are defined by grouping together identical slot positions in successive time frames. Packet networks typically do not allocate fixed resources to transmitters, but rather route packets of data on a best efforts basis, using destination address information contained in packet headers, and network switches and routers. Packet networks are becoming more popular amongst network operators as they often provide better performance, and are more cost effective to install and maintain, than equivalent circuit switched networks. Traditionally, telecommunication networks have made use of time division multiplexed TDM circuits to interconnect network switches or exchanges. However, for the above mentioned reasons of performance and cost, many operators and leased line providers who provide bandwidth to service providers are moving towards replacing TDM circuits with packet networks. However, it is likely that for many years to come, some operators will continue to rely upon TDM circuits to provide all or at least a part of the networks. The carrier network provides leased line services to interconnect first and second customer premises 2,3, both of which make use of TDM transmitters 4,5 to handle multiple information streams. The nature of these streams is unimportant, although they could for example be voice calls, videoconference calls, or data calls. In order to facilitate the interconnection of the TDM streams, the carrier network 1 must emulate appropriate TDM circuits. TDM links are synchronous circuits with a constant transmission bit rate governed by a service clock operating at some predefined frequency. In contrast, in a packet network there is no direct link between the frequency at which packets are sent from an ingress port and the frequency at which they arrive at an egress port. With reference again to FIG. That is to say that the TDM service frequency fservice at the customer premises on the ingress side must be exactly reproduced at the egress of the packet network fregen. The consequence of any long-term mismatch in these frequencies will be that the queue at the egress of the packet network will either fill up or empty, depending upon on whether the regenerated clock fregen is slower or faster than the original clock fservice , causing loss of data and degradation of the service. Also, unless the phase of the original clock fservice is tracked by that of the regenerated clock fregen , the lag in frequency tracking will result in small but nonetheless undesirable changes to the operating level of the queue at the egress. Some reliable method for synchronising both the frequency and phase of the clock at the egress of a packet network to those of the clock at the TDM transmitted must be provided. One approach is to use some algorithm to recover the transmitting clock frequency and phase from timestamps incorporated into packets by the sender, taking into account the transmission delay over the packet network. As the transmission time over the packet network is unpredictable for any given packet, an adaptive algorithm might be used. For example, some form of averaging might be employed to take into account variations in the transmission delay. EP discloses a method of synchronising first and second clocks coupled respectively to ingress and egress interfaces of a packet network, where the first clock determines the bit rate of a constant bit rate TDM stream arriving at the ingress interface and the second clock rate determines the bit rate of a constant bit rate TDM stream sent from the egress interface. The method comprises calculating a minimum packet Transit Time over the network in each of successive time

intervals, and varying the frequency of the second clock so as to maintain a constant value of the calculated minimum packet transit time and hence achieve both phase and frequency synchronisation of the first and second clocks. The underlying principle of the described method is that, almost regardless of the level of traffic in a packet network, a proportion of packets will always be transmitted with the same fixed minimum transit time value. The method is applicable in particular for synchronising a clock of a TDM sending entity coupled to the egress interface, to a clock of a TDM link coupled to the ingress interface. EP is concerned with synchronising the TDM clocks at the ingress and egress of the packet network which are associated with a given TDM stream. The method may be applied in parallel to a number of TDM streams sent between the same ingress and egress interfaces. The clocks and synchronisation procedures of each stream are treated independently. This is especially true where the frequencies with which packets are injected into the packet network at the ingress interface, or arriving at a given network node, are similar for different streams. Consider for example two TDM streams having similar, but slightly different frequencies, i. The packet size is the same for each stream. The top two streams in FIG. Where the TDM streams arrive at different ingress interfaces, the packet flows may intersect at some common, intermediate node within the packet network. At the interface or common node where the packet flows intersect, the forwarding unit is only able to inject one packet at a time into the packet network at the output. It will take packets in sequence as they arrive on the two streams. This is not problematic where packets arriving on the two streams do not overlap. However, when an overlap occurs, the sending of a packet may be delayed, pending the sending of a packet which arrived momentarily earlier for the other stream. This situation is illustrated in the lower sequence of FIG. As an example of this effect, consider the case where two packet streams have a nominal packet rate of 1 khz but are actually 1 ppm offset from each other. The beat period for the disruption is s, and the timing is disrupted for However, very small differences in frequency will have a relatively long-lasting effect on the delays suffered by packets. Assuming that the minimum packet Transit Time method of EP is employed, the correlation between packet flows will result in a change in the minimum transit time determined by the egress interface which is not due to any loss of synchronisation of the ingress and egress clocks, and which is only very slowly varying and therefore not cancelled by the low pass filtering of minimum transit time values carried out at the egress. The egress interface will, wrongly, adjust the egress clock in an attempt to restore the apparent phase and frequency difference between the ingress and egress clocks. The result could be a phase error equal to the magnitude of the disruption, eg. The problem of a loss of synchronisation will arise for other similar clock synchronisation procedures, e. According to a first aspect of the present invention there is provided a method of preparing packets for injection into a packet network at an ingress interface of the packet network, for transmission over the network, the method comprising: Embodiments of the invention result in the significant advantage that the correlation between the packet flows arriving at the sender is reduced. The probability that packets on different flows will overlap for prolonged periods is thereby reduced. While overlaps may still occur, these are likely to be isolated events, or to continue only for a relatively short sequence of packets. The effects of such variations can be removed by low pass filtering or data selection techniques at the egress interface. In a particular embodiment of the present invention, said ingress and egress interfaces are interfaces between the packet network and incoming and outgoing Time Division Multiplex TDM circuits. According to a second aspect of the present invention there is provided a method of preparing packets for injection into a packet network at an ingress interface of the packet network, for transmission over the network to one or more egress interfaces, the method comprising: A number of options for changing the frequency of a packet flow exist. These include, without limitation: Introducing a fixed change in the packet size of a given flow; Dynamically varying the packet size of a given flow. The variation may be random, in order to increase the resilience to prolonged correlation of the packet flow to other packet flows; Changing the packet size according to a pseudo-random sequence; Deliberately introducing a delay into the packets of a packet flow; and Applying a change to the bit rate of a constant bit rate stream. These options may be employed separately or in combination. According to a third aspect of the present invention there is provided a method of

synchronising first and second clocks coupled respectively to ingress and egress interfaces of a packet network, where the first clock determines the bit rate of a constant bit rate stream arriving at the ingress interface and the second clock rate determines the bit rate of a constant bit rate stream sent from the egress interface, the method comprising preparing packets for injection into the packet network at the ingress interface according to the above first or second aspect of the invention. The method preferably comprises, at the egress interface, calculating a minimum packet Transit Time over the network in each of successive time intervals, and varying the frequency of the second clock so as to maintain a constant value of the calculated minimum packet transit time and hence achieve frequency and phase synchronisation of the first and second clocks. According to a fourth aspect of the present invention there is provided apparatus for preparing packets for injection into a packet network at an ingress interface of a packet network for transmission over the network to one or more egress interfaces, the apparatus comprising: According to a fifth aspect of the present invention there is provided apparatus for preparing packets for injection into a packet network at an ingress interface of a packet network for transmission over the network to one or more egress interfaces, the apparatus comprising: However, the rate of packet arrival at the destination interface 7 is perturbed by the intervening packet network. Packets will typically arrive in bursts separated by varying amounts of delay. The delay between successive packets and bursts will vary for example depending on the amount of traffic in the network. The characteristics of the network are non-deterministic, but over the long term the rate of arrival at the destination will equal the rate of departure from the source. At the source interface 6, a timestamp is placed into the header of each packet prior to transmission. If the buffer 10 has zero packets in it when the TDM output requires to transmit, an underrun will occur, which is undesirable. In order to minimise underrun events it is necessary to build up the PDV buffer 10 so that it contains sufficient packets to supply the TDM output for the majority of inter packet delays. However, the PDV buffer 10 cannot be made arbitrarily large because this directly increases the end to end latency which, in general, is required to be as low as possible, the maximum tolerable latency being dependent on the application. For example, voice typically requires lower latency than data. When a packet arrives at the packet input of the destination interface 7, the packet is placed into a queue of the PDV buffer  The Remote Timestamp is extracted from the packet and is passed to a differencer. The destination interface 7 maintains a TDM output counter which is a running total of the bits sent on the outgoing TDM linkâ€”this counter is initialised to the first received Remote Timestamp. A Local Timestamp is obtained for the received packet using this counter, and this is also provided to the differencer. It should be noted that because the source and destination clock frequencies and initial counts i. However, it is true that, given an ideal fixed delay packet network, the Transit Time will decrease if fservice exceeds fregen, will increase if fregen exceeds fservice, and will remain constant if these frequencies are identical. In a packet network, most of the transmission delay is caused by waiting time in queues at output ports of switches and routers. However, a proportion of packets will not be held up in any queues, i. These packets will experience only a minimum delay, the value of which is largely independent of network loading, being due to factors such as cumulative line propagation delays and service delays at each switch. If the network loading varies, the average packet transmission delay over the packet network will also vary. However, the minimum delay should not vary to the same extent. Therefore identifying the minimum packet delay within each of successive time periods should give the required indication of drift between the source and destination clock frequencies, independent of changes in network loading. This is very important where such changes in loading occur at a relatively low frequency, for example a 24 hour cycle. Such low frequency variations may be indistinguishable from source clock frequency drift which must be followed by the clock recovery system. In a typical implementation, for every packet received at the destination interface, a Transit Time is calculated. The minimum Transit Time is reset for each new time period. Immediately after the expiry of a time period, a clock control algorithm will read the minimum Transit Time recorded for that period, determine the correction required to the destination interface clock frequency, and write the required frequency to the DCO of the destination interface. The clock control interval will generally be relatively large compared to the transmission

and arrival intervals between packets so that the minimum Transit Time that the algorithm reads will be the minimum of a large set of Transit Time values. A suitable clock control algorithm is given by the following difference equation: The constants G1 and G2 determine the frequency response of the system and are selected to track long term drift in fservice but reject short-term variation due to packet delay variations. A further term may optionally be added to the above equation. This makes use of an Offset constant which can be used during operation to adjust the operating point i. This may be desirable in order to cope with changing network conditions which cause the buffer to empty or overflow. A filter function, such as a first order filter, may be used to provide a filtered measurement of the PDV buffer fill level. The clock control algorithm can then be expanded to read the filtered level, and set the Offset accordingly. This system is robust in the presence of lost packets because the Remote and Local Timestamps of the next packet received following any lost packet s are unaffected by the loss. The lost packets merely represent a short term loss of resolution in the measurement.