The Synchronization of Time-slotted Photonic Star Networks

By

Abdalla El Shazli

Electrical and Computer Engineering
McGill University, Montreal

August 2006

A thesis submitted to McGill University
in partial fulfillment of the requirements of the degree of
Master of Engineering

Your file Votre référence
ISBN: 978-0-494-32587-2
Our file Notre référence
ISBN: 978-0-494-32587-2

Abstract

Photonics is a technology capable of supporting very high bit-rates of data. However, with the current state of photonic technologies logic and memory functions are very difficult to implement in the photonic domain.

In photonic star networks using time division multiplexing (such as the Agile All Photonic Network), timeslots from the edges of the network have to arrive at the star point at exactly the same instant to be switched because they cannot be buffered in the photonic domain. The switching requires that the time at which the timeslots are transmitted must be coordinated and tightly controlled.

This thesis addresses methods of synchronizing the components at the edge of the network to compensate for heterogeneous propagation delays between the edges and the star point. Different methods of providing this compensation are described and assessed in terms of their capabilities and performance.

Abstract (French)

La théorie de la photonique suggère qu'il possible d'atteindre des débits binaire de données très élevés. Cependant, il est encore très difficile d'implanter dans le domaine photonique la logique et la mémoire rendant possible ces hauts débits à l'aide des technologies disponibles à ce jour. Dans les réseaux photoniques en étoile qui utilisent le multiplexage par répartition dans le temps (tel que les Réseaux Agiles Tout Photoniques), les intervalles de temps provenant de la périphérie du réseau doivent arriver à l'organe central simultanément pour être commutés car la mise en mémoire-tampon est impossible dans le domaine photonique. Le moment auquel les intervalles de temps sont transmis joue un rôle critique dans la commutation et requiert un contrôle accru. Ce mémoire de maîtrise présente le potentiel ainsi que la performance de diverses méthodes de synchronisation utilisées pour compenser pour l'inégalité des temps de propagation entre la périphérie et l'organe central du réseau de chacun des composants.

Acknowledgements

First of all, I would like to express my unlimited gratitude to God for all what I achieved. Second I would like to express my gratitude to my supervisor Professor Richard Vickers for without his inspiration, patience and vast knowledge and experience this thesis would not have been possible. I owe a great deal to my supervisor Professor Richard Vickers for his patience and confidence in me, which have been truly amazing and I will be indebted to him for a long time because of it.

Thanks also to my parents and family, who provided the item of greatest worth - opportunity. Thank you for standing by me through the many trials and decisions of my educational career. My father deserves special thanks for making me believe that I could always do whatever I put my mind to as long as I worked hard enough. Thank you my brother Ahmed for helping me whenever I needed your help.

Finally, I would like to thank all my friends and especially Osama Abdel-Mannan, Ali Haidar, Amer Shaban, Imad Khazali and Mahdi Hermas for not only helping me in technical matters, but for their friendship and the good times we shared.

Abdalla El Shazli

McGill University,
August 2006

Table of Contents

List of Figures

1 Introduction

This thesis addresses the synchronization of the components in a star network with a time-slotted photonic data path. The synchronization is required to prevent collisions within the network with the subsequent loss of data.

1.1 Background

Early telecommunication networks were simple in structure but as the number of users and the geographical span of networks have increased, the complexity has also increased. The increased network complexity is a consequence of the limited capacity of transmission and switching components. In the 1980s, the introduction of digital switching technology and optical fiber transmission allowed both the aggregation of switching and transmission facilities. Digital switching technology allowed for an increase of the number of telephone circuits and optical fiber transmission allowed for the aggregation of transmission circuits. This resulted in a reduction in the number of switches, and the simplification of the underlying transmission architecture by the use of SONET/SDH rings. Recently, further gains in switching capacity have been made with the use of ATM and then IP technology, and dramatic increases in the capacity of optical fiber transmission systems so that current backbone network architectures are based on high capacity optical transmission, with electronic switching points. Background references on communications and specifically digital communications are in [22][23][24][25].

Electronic switches are limited in capacity by factors such as physical size, power consumption, and the speed of the technology. The next stage in network development will require photonic switches. Photonic switches are available, but compared to electronic switches have some limitations. First, photonic logic is not available in the complexity, power consumption or cost of electronics so the control of photonic switches will continue to be performed by electronics. Second, there is very limited capability to store information in a comparable manner to Random Access Memory (RAM) once it is in the photonic domain. The only practical storage is in the form of delay lines, which are very limited in application. Third, current commercially available photonic switching devices have slow switching speeds and limited port counts. They can be used for switching the light in complete fibers, or individual wavelengths. The switching mode is circuit switching, and although applicable to the switching of bulk or aggregated traffic some form of switching of smaller units of information is required before the full benefits of a photonic network can be realized. Two switching techniques for smaller units of information are the subject of research: Burst Switching and Timeslot switching. The limitations of photonic technology means that both techniques use out-of-band signaling and control, and both require electronic buffers at the ingress point prior to the conversion of the data into a photonic form for transmission.

Burst Switching uses variable length data units. There are various forms of Burst Switching, but most result in a relatively high probability of collision and therefore data loss. The minimization and management of loss is the subject of ongoing research in this area. Timeslot switching uses scheduled arrivals at the switch to avoid data collisions. However, for a scheduled switch to operate without loss the information has to arrive at the switch at the exact instant in time it has been allocated for switching. The time at which the information is sent from the edge of the network has to be offset by the propagation delay and equalized for all edges of the network. A process for calculating and maintaining this offset is what is described in this thesis.

In an electrical star network data units may be buffered at the central node before switching, and therefore time synchronization of the individual stations is not necessary.

Where the data path between the stations is a light path buffering at the central node is not possible with the current state of photonic technology. If the distance of the stations from the central node is large with respect to the size of data units then some form of time coordination is required to ensure that collisions do not occur. The Agile All-Photonic Network (AAPN) [1] is an example of a photonic star network. It is intended to be applied across a large geographical area where the propagation delays from the Edge Switches to the Core Switch can range from 0 to several tens of milliseconds. The data unit used for communication across AAPN is a fixed duration timeslot (e.g. 10μsec).

2 Thesis Objective and Organization

This chapter presents the research objectives and the thesis organization. First, the goals of the research are stated and described in detail. This is followed by a description of the organization of the thesis and the contents of each chapter.

2.1 Thesis Goals

The main goal of this thesis is to present and compare solutions for time synchronization in time-slotted photonic networks. The specifics of the network and of the problem of time synchronization will be explained in the next chapter. The thesis will demonstrate that existing methods of synchronization are not applicable and it will compare three possible solutions. The goal is not to recommend a specific solution for a particular network problem but to provide information and a methodology for network designers to improve the synchronization method. In this thesis, the various types of clocks and their characteristics will be described, and how they may be applied in the design of the synchronization system. Performance of the synchronization system will be assessed using system modeling. Actual implementation was not considered because of the cost and complexity. Modeling should provide sufficient information for implementers.

2.2 Thesis Organization

The remainder of this thesis is divided into six chapters. The third chapter provides a clear description of the network and components for which synchronization is required and the performance bounds required for the system. The fourth chapter describes research into the different types of synchronization and clocks suitable for application in the network. The fifth chapter discusses in details synchronization methods. The sixth chapter includes a performance analysis. Finally, the last chapter presents the general conclusions.

3 Problem Definition

An outline of the problem was given in the introduction. Details of the network and components will be given in this chapter. The following patents give a strong background of understanding the problem and AAPN [3][4][5][6].

3.1 Agile All-Photonic Network (AAPN)

Although the solutions presented in this thesis are applicable to any time-slotted photonic star network or in an extended form to any scheduled access photonic network the Agile All-Photonic Network will be used as an example of such a network The AAPN provides a focus for many research topics across a number of universities.

The following definitions will be used in the description:

- Ingress Edge Switch – the component that collects data from various sources in electronic form, assembles it into timeslots, buffers the timeslots and at the instant they are transmitted transforms them into photonic timeslots.

- Ingress Selector – a component which aggregates timeslots from multiple Edge Switches by selecting a single input for the duration of a timeslot.

- Core Switch – A photonic switch that opens photonic paths between input ports and output ports for one timeslot duration. During subsequent timeslots the configuration of the switch will generally change. Each output port can be considered as providing an identical function to an Ingress Selector. The data on the output port consists of a stream of timeslots from different Ingress Edge Switches separated by a small guard interval.

- Egress Distributor – A photonic switch that distributes the timeslots on a single wavelength to multiple Egress Edge Switches.

- Egress Edge Switch – A switch that receives timeslots from multiple Core Switches, recovers data and restores it to its original format.

- Edge Node - consists of Ingress and Egress Edge Switches together with their control systems

- Core Node – consists of a Core Switch together with the control for the switch and the master control for the Edge Nodes. The timing source for the network will be located at the Core Node.

- Selector – consists of an ingress selector and a distributor together with the control

Assumptions:

Data buffers cannot be implemented in the photonic domain

Differences in propagation delay between different wavelengths on the same fiber are negligible. If this assumption were incorrect the solutions presented would only be valid for a single wavelength.

A control channel exists between the Core Node and the other components. The exact form of this control channel is not described, but as control is electronic the channel must be terminated in electronic form. Collocation of Edge Nodes with Selectors and Core Nodes is one means of converting the control channel to an electronic form.

The incoming side of the photonic switch does not have a means of detecting an optical control channel. This implies that the control channel must traverse the switch prior to conversion to electrical

The data path structure of AAPN is illustrated in Figure 3-2. From the ingress to egress the data can traverse ingress Edge Switch, a Selector, A Core Switch, an Egress selector and an Egress Edge Switch. The data unit that passes across AAPN is a Timeslot. The content of the timeslot is not relevant to the synchronization issue, but may consist of data from a number of sources such as IP routers, Telephone switches, ATM switches etc. The bit rate of the information in the timeslot does not directly impact the synchronization scheme, but would typically be of the order of 10Gbit/s.

Timeslots originated by an Ingress Edge Switch are sent at fixed intervals (Figure 3-1). The intervals at which they are sent include a Guard Period. The Guard Period allows for a finite switching time, variations in timing and other effects that influence the boundaries of the timeslots. It is sufficiently long to allow these variations, but not so long as to significantly affect the utilization of the transmission facility.

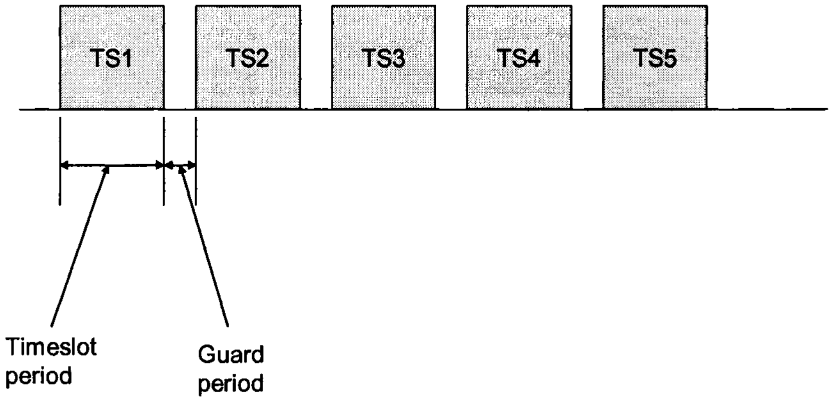


Figure 3-1 – Arrangement of Timeslots originating from Edge Switch

3.1.1 AAPN Data Path

An AAPN (Agile All Photonic Network) is implemented with Core Nodes, Edge Nodes and Selectors (Figure 3-2).

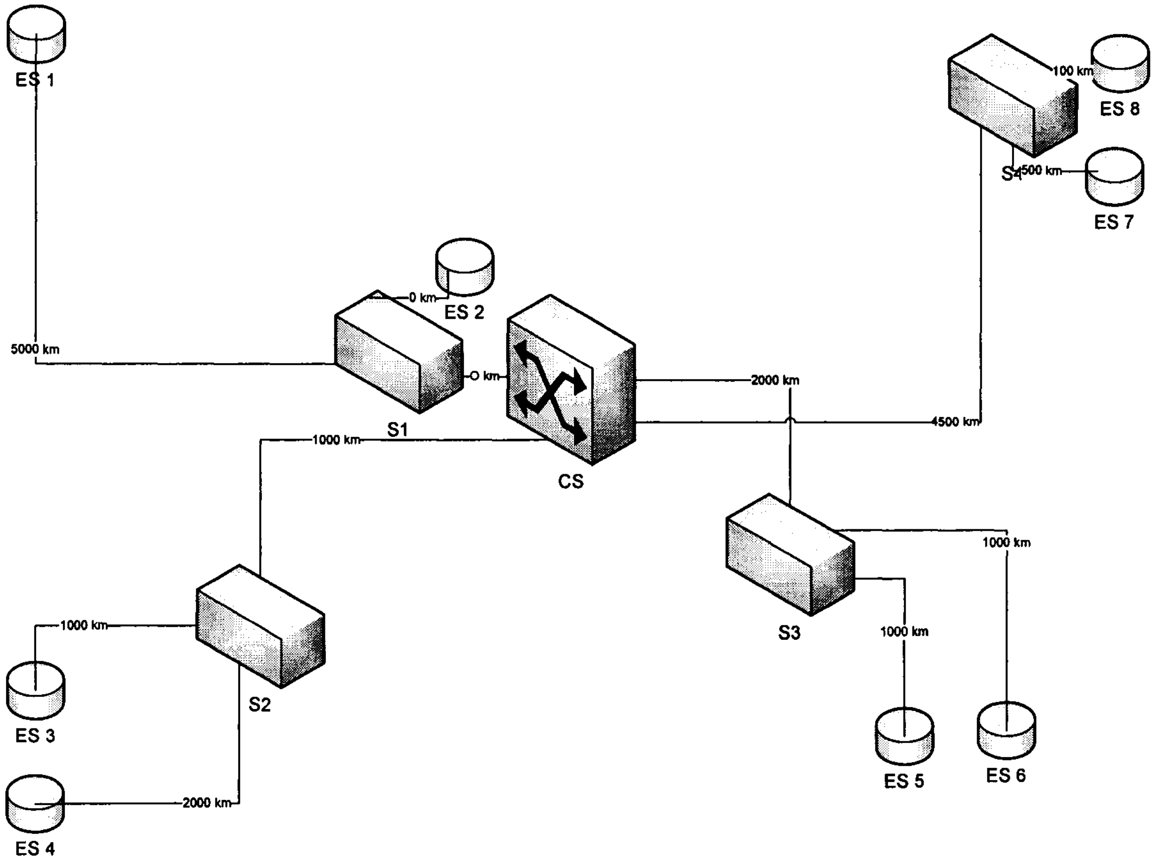


Figure 3-2 - Agile All-Photonic Network Topology

Figure 3-3 is intended to illustrate the operation of the Core Switch. On the input side there are four input ports each with a stream of timeslots. The shaded timeslots are to be switched to the same output port. The un-shaded timeslots may be unused, or may be switched to other output ports. The configuration of the switch in each timeslot is set by the Switch Control. The Core Switch is a space switch so TS1 fills exactly the same space in time as TS1 on the input. The timeslots on the input ports must be exactly aligned to maintain the timing on the output side. A minor misalignment (less than the Guard Period) will result in erosion of the Guard Period on the output port, but a major misalignment will cause clipping of the misaligned timeslot with the resultant loss of data. In Figure 3-4 Input port 4 is delayed with respect to the timing on the Core Switch. The first part of the timeslot will be passed, but by the time the last part of the timeslot reaches the switch the path will be set from Input port 3 to the output port, and therefore the end of the timeslot from Input port 4 will be lost. Once the stream of timeslots has been assembled at the output port there is no requirement for synchronization at the Egress Selector or the Egress Switch as the data in the timeslots can be extracted asynchronously and there is no potential for collision.

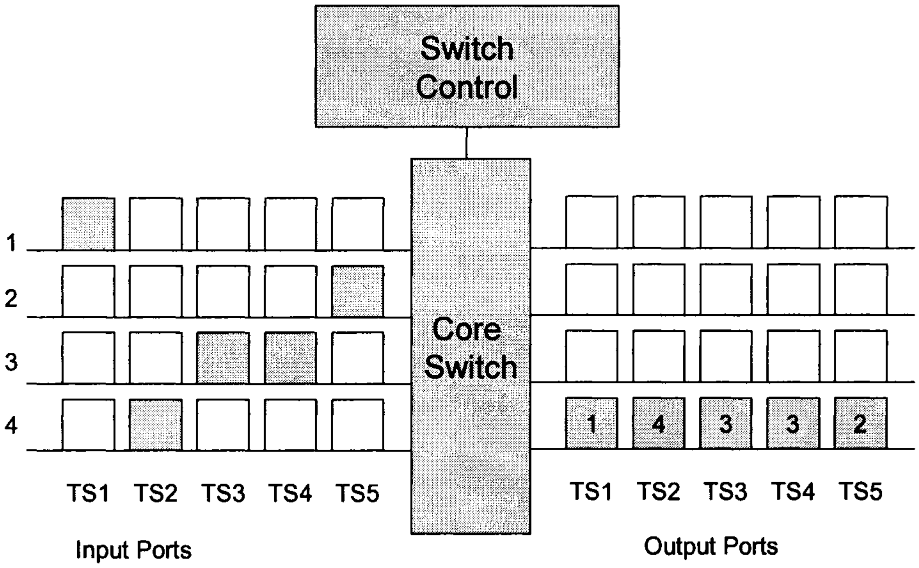


Figure 3-3 – Operation of Photonic Switch

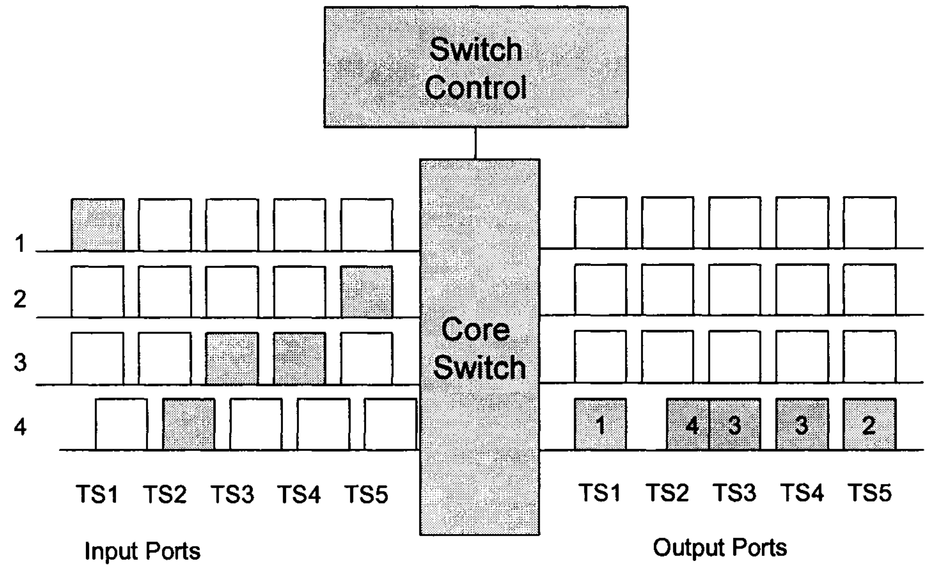


Figure 3-4 – Operation of the Core Switch with a Misaligned Input Port

If the delays from the Edge to the Core are small compared to the timeslot and guard period, or the delays from all Edge Switches are identical then the misalignment will not be significant. In AAPN though the delay from Edge to Core can be from zero up to tens of milliseconds. In an electronic switch the information could be buffered at the Core Switch to equalize delays, but in the absence of equivalent photonic technology buffering is not an option. The method proposed to align the timeslots at the Core Switch will now be briefly described.

A master clock is maintained at the Core Node. Each Edge Node contains a clock that is offset (advanced) by the propagation delay between the Edge Switch and the Core Switch. In this way when the Edge Switch sends a timeslot at a given time as indicated by its clock, the timeslot will arrive at the Core Switch at exactly the same time according to its master clock. The timing is complicated by the presence of the Selector. A selector is equivalent to a switch with multiple input ports and a single output port. In networks with Selectors the timeslots from the Edge Switches have to arrive at exactly the same instant, and the timeslots from the Selectors have to arrive at exactly the same instant at the Core Switch. This process is illustrated with the following example:

If Edge Switch1 wants to send data to Edge Switch2 then the packet should be received by the Core Switch at the time slot assigned for such communication, for example timeslot 4. Therefore Edge Switch1 must send the packet such that it is received by the Core Switch at timeslot 4. To make this happen Edge Switch1 must be synchronized with the Core Switch in synchronization phase such that it knows how long it will take the packet to reach and when it should be sent. In this scenario the Edge Switch will know how long it takes to the selector and then the latter should also know how long it takes to the Core Switch. Therefore two levels of synchronization are required, one for the Edge Switch and one for the selectors.

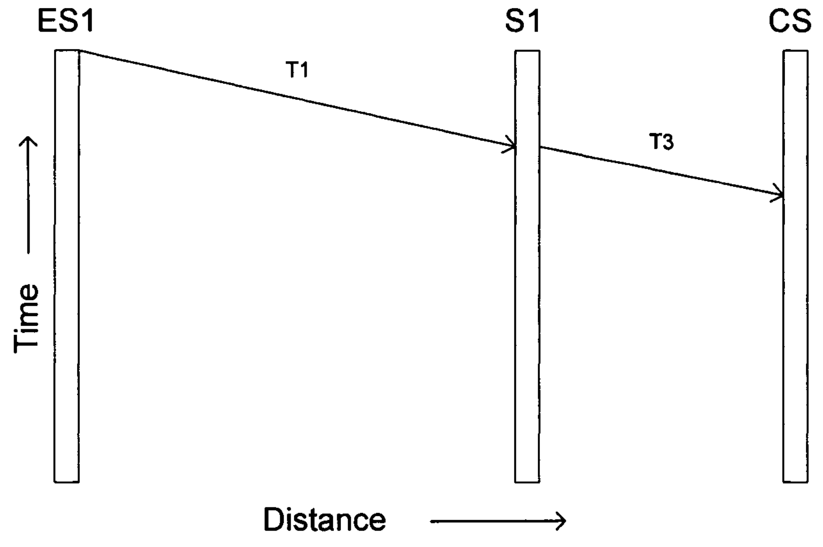


Figure 3-5 - Synchronization of one Edge Switch with Core Switch

As explained previously Time Division Multiplexing (TDM) is used in AAPN. An algorithm has been implemented to provide connectivity of all Edge Switches to the Core Switch and to each other using timeslots. In AAPN it is a unidirectional connectivity, meaning that at certain timeslots there is an input port and an output port. In the following example the rows and columns are the ports for the Core Switch. The row represents the output, while the columns represent the input. There are up to 64 x 64 ports but an 8 x 8 switch is used for illustration (Table 3-1):

	1	2	3	4	5	6	7	8
1					X			
2						X		
3							X	
4								X
5		X						
6				X				
7	X							
8			X					

Table 3-1 – Crosspoint Selection in 8 x 8 Switch

Assuming that port 1 is for the Core Switch control and synchronization purposes. The other ports 2 - 8 are to Edge Switch 2 - 8 respectively. The x mark indicates that this link or connection is valid. Therefore in this case at this timeslot, links are established as follows:

- Port 7 sends to Port 1 (Core Switch)
- Port 5 sends to Port 2
- Port 8 sends to Port 3
- Port 6 sends to Port 4
- Port 1 (Core Switch) sends to Port 5
- Port 2 sends to Port 6
- Port 3 sends to Port 7
- Port 4 sends to Port 8

During this timeslot connections should be made between the ports as shown. Therefore if Edge Switch 5 wants to send packets to Edge Switch 2, then Edge Switch 5 should send during this timeslot. For this to happen Edge Switch 5 should send its packets so that they reach the Core Switch at this timeslot so that communication can take place. The scheduling algorithm to provide the timeslot timetable is outside the scope of this thesis but the solution of making the Edge Switch and Core Switch send and receive packets will be addressed. Showing multiple timeslots gives a better point of reference.

The Core Switch will be investigated and seven successive timeslots are used. Table 3-2 shows the ports that send to the Core Switch for synchronization and control purposes and each timeslot.

Port 3	Port 2	Port 7	Port 5	Port 4	Port 6	Port 8
TS 1	TS 2	TS 3	TS 4	TS 5	TS 6	TS 7

Table 3-2 – Control Timeslots

The above table shows that each timeslot is assigned to a port or an edge switch. In this case if timeslot 4 is taken, Edge Switch 5 will send Core Switch packets. Edge Switch 5 then knows exactly when to send its packet to the core switch so that they arrive at the beginning of the 4th timeslot. This does not happen if Edge Switch 5 does not have the exact clock (time) it needs to send the packet to the Core Switch. Suppose that timeslot 4 starts at time = 5.39s and assume that the propagation delay between Edge Switch 5 and the Core Switch is 0.39 s. Therefore Edge Switch 5 should send its packet at time = 5.00s so that it arrives at the Core Switch at the start of the 4th timeslot.

If at timeslot x Edge Switch1 sends Edge Switch2 and Edge Switch3 sends Edge Switch4. The following scenario should occur:

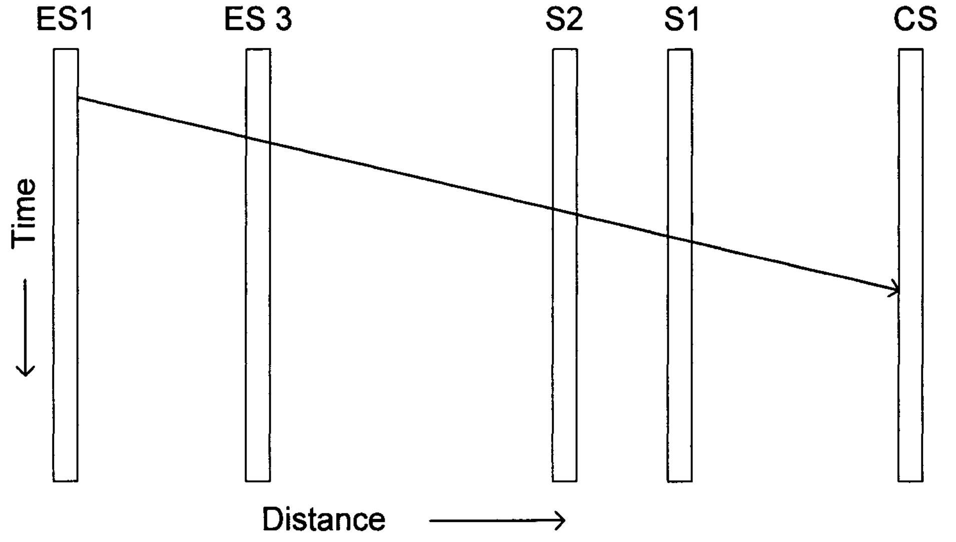


Figure 3-6 – Synchronization of two Edge Switches with the Core Switch

In Figure 3-6 Edge Switchl should send its packet before Edge Switch3 so that both packets are received by selector 1 and selector 2 respectively. Moreover selector 2 should send before selector 1 so that both packets arrive at the Core Switch at the exact time. At timeslot x+1 (Figure 3-7) Edge Switch2 sends to Edge Switchl and Edge Switch4 sends to Edge Switch3, the following will occur:

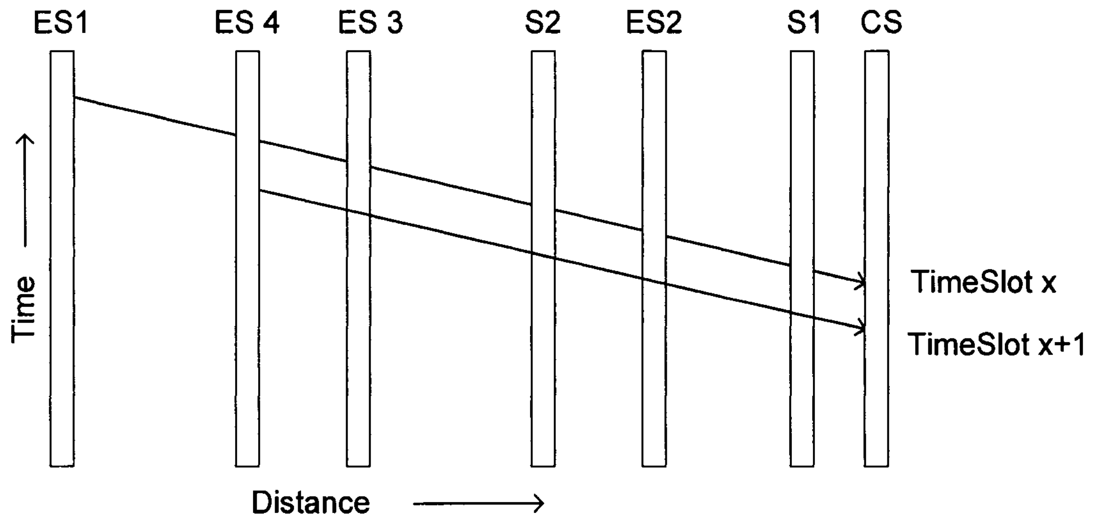


Figure 3-7 – Timeslot Transmission (timeslot x+1)

This thesis will outline how these schemes and timing mechanisms can be synchronized with the Core Switch. Once the initial synchronization is achieved the alignment of the clocks has to be maintained in the presence of clock drift, variable temperatures and other effects that create timing errors. For example all Edge Nodes and Selectors are already synchronized with the Core Switch. Communication already takes place between the Edge Switch and the Core Switch. Depending on the type of clocks used in the Edge Switch, Selectors and Core Switch will drift. This means that clocks which were exact and synchronized will be a bit different.

A numerical example illustrates: Core Switch clock is at t=0, depending on the scheme used for synchronization and other clocks might be the same or offset. In this example it is assumed that all selector and Edge Switch clocks will also be at t=0. After some time passes the clock for Edge Switch 1 drifts by 0.001 s. Therefore instead of t= 5.00 s for this Edge Switch it is t=5.001s which causes a loss in synchronization. This problem requires a maintenance process or solution.

3.1.2 Control Architecture

The physical manifestation of the control channel in AAPN has not been studied, but it is necessary to detail the requirements for the control channel and present an example of how it may be implemented. Assuming there is a well-known wavelength on which all control information is carried and that the wavelength is divided into timeslots and user data may also be carried in timeslots on the same wavelength. Timeslots carrying control data carry only control data. A diagram is given in Figure 3-8. Note that this configuration only applies on the wavelength carrying the control channel. All other wavelengths in use only carry user data.

At the Core Switch one egress port and one ingress port on the well-known wavelength are dedicated to signaling and Optical-Electronic and Electronic-Optical interfaces are associated with the port. At the Selector, means must be provided for terminating control channels to and from the associated Edge Nodes as well as the Core Node. At the Ingress Selector one input port may be allocated to control for the insertion of control messages destined for the Core Node. On the output side of the selector a means must be provided for receiving control messages from the Edge Nodes. This may be achieved by passively splitting the output signal and reading the control timeslots, or by placing a 1x2 switch with one output directed to Selector control and the other providing the connection to the Core Switch. At the Egress Distributor one output port may be dedicated to control, and some means provided on the upstream side to insert messages destined for the Edge Nodes. As with the Ingress Selector insertion may be achieved passively with a coupler, or with a 1x2 switch.

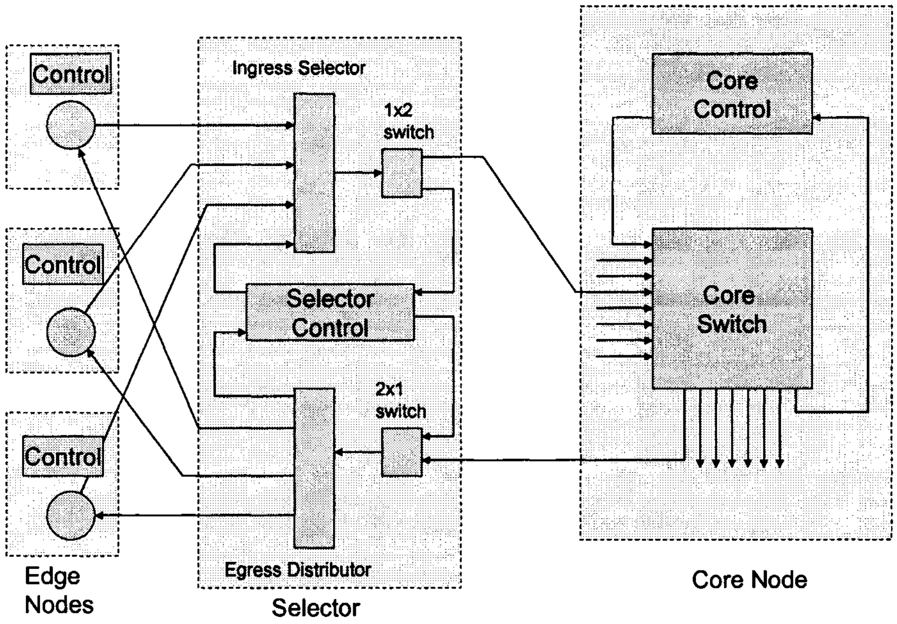


Figure 3-8 – Example Control Architecture

3.1.3 Synchronization Process

The synchronization process is the establishment and maintenance of the timing offsets described in Section 3.1.1. It is part of the Control Plane, and uses the communications provided by the Control Channel described in Section 1.1.1. When the network is first activated, or when a new component (Edge Node or Selector) is added there is no way of aligning the clocks at the Edge Node, Selector and Core Node without some form of handshaking protocol between them. This initialization protocol is to set the offset of the Selector/Edge Node clocks with respect to the Core Node master clock to equalize propagation delays from different components (Figure 3-9). Three methods of achieving this initial synchronization will be described. Once the system is aligned correctly clock drift and propagation delay changes can cause the alignment to drift, and an additional protocol is needed to maintain the alignment within a certain tolerance. Some types of underlying physical network (e.g. ring networks) have protection schemes that will restore a failed facility over a geographically different route. If the propagation delays on the new route the alignment will be lost. This loss of alignment has to be detected and corrected before traffic can be sent over the new route. A method will be described of detecting and correcting this loss of alignment.

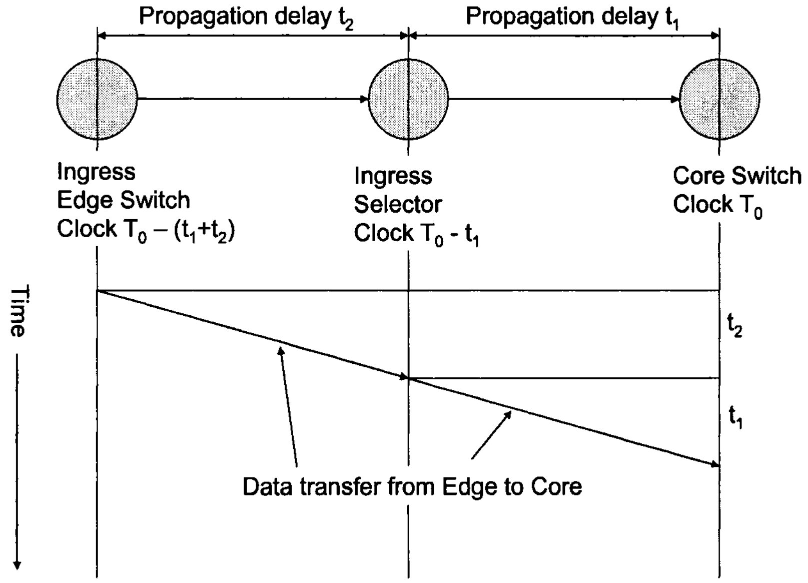


Figure 3-9 – Alignment of Clocks

4 Literature Review

4.1 Synchronization

The English definition of synchronization is to "cause to occur or operate at the same time or rate" (Oxford Dictionary). In digital systems many tasks and operations follow a specific procedure and the synchronization process is responsible for making sure that the operations happen in the correct order and sequence. Synchronization is not only found at the hardware level to synchronize carriers, bits, frames and packets but it is also found in the software to ensure that the correct sequence of messages are sent and received. Synchronization also occurs on a larger scale at the network level. This process ensures that an entire network is aligned and communicating properly.

To ensure that synchronization is implemented correctly, networks require a protocol. A Protocol is "a set of rules governing the exchange of transmission of data between devices" (Oxford Dictionary). Structured as layered models, most communication protocols are software oriented. In layered models the lower layers provide the upper layers with the required services and information. For example, the Open System Interconnection (OSI) model divides the network into 7 layers: the Application Layer, Session Layer, Presentation Layer, Transport Layer, Network Layer, Data-Link Layer, and the Physical Layer. The Application, Session and the Presentation layers can be combined into one layer and are generally implemented in the end-stations. Each layer has different synchronization techniques for achieving its unique role. For instance, at the physical layer there is carrier synchronization, bit synchronization and symbol synchronization. This thesis will demonstrate the different types of synchronization and it will explain which type is needed for the problem and how it is related. Most of the information in this section is taken from Bregini [2].

4.1.1 Carrier Synchronization

"In amplitude modulation (AM) systems, modulation is given by multiplying the modulating signal $s(t)$, usually with zero mean, by the carrier wave $\cos 2\pi f_0 t$ in the form

$x(t) = s(t) \cdot \cos 2\pi f_0 t$ (1.1) or

$[1 + ms(t)] \cdot \cos 2\pi f_0 t$ (1.2)

In the latter case, the envelope of the modulated signal $x(t)$ is proportional to $s(t)$ if $|ms(t)| \leq 1$" (Bregini [2], p.4). This permits an easier design of demodulators (*envelope demodulation*).

In the former case demodulation is made possible by multiplying the modulated signal by a single wave with the same frequency and phase of the carrier (*coherent demodulation*) followed by low-pass filtering, since

$x(t) \cdot \cos \omega_0 t = s(t) \cdot \cos2\ \omega_0 t = s(t)/2*((1 + \cos 2\omega_0 t)$ (1.3)

This type of demodulation requires that the multiplier signal $\cos\omega_0 t$ that was used in the receiver (carrier reconstructed) has *exactly* the same frequency and phase of the modulated carrier as received. Any phase shift β yields an attenuated signal $(s(t)/2) \cos \beta$ at the output of the low-pass filter (a null signal if $\beta = \pi/2$!).

These formulas indicate that amplitude modulation is equivalent in the frequency domain to a translation of the modulating signal spectrum to the carrier frequency $f0$, in cases where the carrier itself is added. The spectrum of an AM signal is redundant since the two mirror parts around the carrier frequency f_0. Single Side Band (SSB) modulation consists of transmitting only one of the two mirror halves, which achieves higher bandwidth efficiency. SSB demodulation must be coherent; here, the phase alignment of the carrier reconstructed is even more critical, as any phase shift also yields a distortion of the demodulated signal. Coherent demodulation is based on carrier reconstruction, which relies on the recovery of a signal coherent with the carrier in frequency and phase.

Carrier reconstruction is simple if in the received signal spectrum there is a line at the carrier frequency f_0 as it happens and the modulating signal does not have a mean of zero. In this case, carrier extraction may be accomplished by means of a suitable narrow passive band-pass filter or a Phase-Locked Loop (PLL) [20]. The loop is designed with a narrow band to allow the Voltage Controlled Oscillator to lock and track small frequency fluctuations around the nominal frequency f_0. In many cases the spectrum line at the carrier frequency f_0 is absent [21]. From the information-transmission point-of-view this is more effective because the power of the f_0 line is wasted with respect to the signal-to-noise power ratio. It also implies that there is a need for a more sophisticated synchronization system capable of reconstructing the carrier in phase and frequency.

A simple example of carrier synchronization is Binary Phase Shift Keying (BPSK) digital transmission, where the symbols 1 and 0 are independent, have the same probability and are coded with antipodal square pulses. The modulated wave is of the form $\pm\cos\omega_0 t$ and the power spectrum is continuous without discrete lines at frequency f_0. Only non-linear transformations can generate the desired f_0 line from the received signal. Squaring and frequency-dividing solves this problem: squaring the modulated wave deletes modulation and yields $(1 + \cos 2\omega_0 t)/2$, generating the line at frequency $2f_0$ from which the desired carrier can be extracted by frequency division. If a quaternary phase modulation (QPSK system, with transmission of 2-bits symbols) is considered, the synchronizer will be based on raising the signal to the 4th power to delete the modulation and then generate line at frequency $4f_0$.

4.1.2 Frame Synchronization

The next step, after the lower layer (physical layer) carrier synchronization has been accomplished and the logical information has been extracted from the incoming signal, is at a higher layer (data link) of abstraction. The goal at the layer of *frame synchronization* is to determine the start and the end of code words or of groups of code words, i.e. to delineate the frames in the raw undifferentiated stream of received bits. In digital transmission, bits are often organized in frames to assign different meanings to the bytes transmitted [2]. Bytes at different positions within a certain frame may be assigned to different user channels sharing the same physical medium in Time Division Multiplexing (TDM). For example, in the European PCM primary multiplex, there are 30 channels allocated to different telephone calls, or also assigned to overhead functions (error check, transport of management and control information.

Frame synchronization is an integral step in digital transmission. The *semantics* itself of the received bits, and thus e.g. proper de-multiplexing of tributaries begins from the correct delineation of frames [2]. All frame synchronization strategies (also referred to as *frame alignment* strategy) are made of three basic operations: 1) *hunting*, whenever the equipment (*frame aligner*) is out of frame synchronization; and the frame alignment pattern is sought in the received bit stream and 2) *maintenance*, whenever the equipment assumes to be frame-aligned and continuously checks frame boundaries where these are supposed to be. 3) *Frame Loss*, is deciding that a previously aligned signal has lost alignment

Frame synchronization is assisted by an *alignment word*, set at special values, which is usually at the beginning of frames. Hunting is performed by seeking the alignment word pattern in any position of the received bit stream. Maintenance is done by checking that a word occurs where the frame aligner supposes that frames begin. Unfortunately, during hunting the alignment word pattern sought can be simulated in any position by the data bit stream, while during maintenance the alignment word, checked where it really is, cannot be recognized owing to line errors. Research has addressed this issue, although it is outside the scope of this thesis. This thesis will demonstrate that time-slot synchronization is another type of frame synchronization that uses time slots rather to synchronize rather than frames. Similar methods of synchronization will be explained later on in the thesis.

4.1.3 Network Synchronization

Network synchronization is the distribution of time and frequency over a network of clocks, spread over a wider geographical area [16]. The goal is to align the time and frequency scales of all the clocks, by using the links interconnecting them (e.g. copper cables, fibre optics, and radio links) [8]. Network synchronization plays a central role in digital telecommunications, having a determining influence on the quality of most services offered by the network operator to its customers. In general, the following are some of the better known applications:

- Synchronization of clocks located at different multiplexing and switching points in a digital telecommunications network.

- Synchronization of clocks in telecommunications networks which require some form of time-division multiplexing multiple access, such as satellite networks, GSM (Global System for Mobile communication) mobile terminals.

4.1.4 Synchronization of Real Time Clocks

The distribution of a reference of absolute time (as the national standard time) to the equipment real-time clocks of a telecommunications network (*synchronization of real-time clocks*) is a different kind of network synchronization. A national standard time aids network control and management. Synchronization of real-time clocks, in particular, distributes the *absolute time* information (e.g.: '23 Dec 1998, 01.32.04 AM', or any other kind of time stamp) and poses different requirements for accuracy. For management needs, a time accuracy of a 100 ns is adequate. Regardless of the start phase offset, the goal of network synchronization is to minimize time error fluctuations among the clocks. This implies that *synchronous physical timing signals* (e.g. sine waves) are distributed to the network clocks by suitable means. Synchronization of digital telecommunications networks achieves time deviations not greater than 10 ns or 100 ns.

The synchronization of real-time clocks, is often accomplished by exchanging messages that carry the time information (time stamps), according to a suitable protocol, along the communication links between the network nodes. A Network Time Protocol (NTP) is an example of this [7]. NTP is used by Internet time servers and their clients to synchronize real-time clocks, as well as to automatically organize and maintain the time synchronization subnet itself. It is designed for high accuracy, stability and reliability even when used over typical Internet paths involving multiple gateways and unreliable networks [9].

The protocol is based on messages transported over Internet Protocol (IP) and User Datagram Protocol (UDP) packets [8]. That provides a connectionless transport service, although it is readily adaptable to other protocol suites. Optional features include message authentication and encryption, as well as provisions for remote control and monitoring. In NTP one or more primary servers synchronize directly to external reference sources. Secondary time servers synchronize the primary servers according to a suitable hierarchy. The reconfiguration of alternative synchronization routes makes it possible to survive outages and failures. The algorithm estimates and compensates for the random transport delay of packets across the network to achieve absolute time accuracy within a few milliseconds.

4.1.5 Summary

This thesis will concentrate on Network synchronization and it will also tackle aspects of the synchronization of real time clocks. For many digital communications engineers, the term *synchronization* is familiar, in terms of the acquisition and tracking of a clock in a receiver, with reference to the periodic timing information contained in the received signal. However, in several areas of telecommunications synchronization plays an essential role, at both different levels of abstraction and contexts. At different abstraction levels, the main contexts in which the word synchronization is used in telecommunications are the following:

- *carrier synchronization*, i.e. the extraction of the carrier from a modulated signal in coherent demodulation;

- *frame synchronization*, i.e. the identification of the start and end of code words or of groups of code words (frames), or also the delineation of the frames in the raw and undifferentiated stream of received bits;

- *packet synchronization*, i.e. the delay equalization of packet arrival times in order to reconstruct a user circuit with constant bit rate over a packet-switched network;

- *network synchronization*, i.e. the distribution of a common timing over a network of clocks, spread over an even wider geographical area;

- *synchronization of real-time clocks*, i.e. a substantially different kind of network synchronization in which the distribution of the absolute time (e.g. the national standard time) across a telecommunications network is concerned, mainly to network management purposes.

4.2 Clocks

4.2.1 Introduction

In theory, the main principle of a clock is to generate periodic oscillations, which is then translated to time. Originally clocks measured time like the Egyptian obelisks and sundials, but as technology developed the types of clocks and the ways of measuring advanced. The main part of the clock is the oscillator, which refers to any kind of periodic physical phenomena. An example of this could be the swinging of a pendulum, a wheel in mechanical clocks, the vibration of atoms in a crystal around their minimum-energy position in quartz clocks, or the radiation associated with specific quantum atomic transitions in atomic clocks [2].

Primarily there are two types of clocks: frequency sources and time sources. A frequency source supplies a (pseudo)-periodic signal whose meaningful information is frequency, which is used for frequency synchronization. Frequency sources do not deliver either UTC or any other time scale. A time source does supply a signal that carries absolute or relative time information. The most remarkable time sources are the UTC sources.

It is also necessary to distinguish between primary and secondary frequency standards. A primary frequency standard is a source of frequency that is a signal usable as reference for frequency synchronization or for other metrology purposes, which does not need to be guided by any external reference. Examples of primary frequency standards are the cesium-beam and hydrogen-MASER atomic standards. A secondary frequency standard is a source of standard frequency that can be, and usually is, guided by a primary standard source. Examples of secondary frequency standards are the high-quality quartz-crystal oscillators and the rubidium-gas-cell atomic standard.

This chapter will discuss three types of clocks: 1) the quartz-crystal oscillators; 2) atomic clocks and; 3) GPS clocks. The research of this chapter was done and most information was taken from Bregini [2], Guinot [15] Major [1] and many others. Tables however are taken from Bregini [2].

4.2.2 Quartz-Crystal Oscillators

Quartz-crystal oscillators are based on piezoelectricity and are present in almost all kinds of modern telecommunications systems. For example, cheap crystal oscillators supply timing to both electronic boards and wristwatches. Stand-alone high-quality crystal oscillators are used as secondary frequency standards. In atomic standards they are also the output frequency source where they are steered by the atomic resonance.

4.2.2.1 Plain Crystal Oscillators

A Crystal Oscillator (XO) is an electronic oscillator while quartz oscillators are based on the piezoelectric effect. The former is where a quartz crystal is excited by a periodic electrical signal at the resonance frequency (in the range from 10 kHz to 1 GHz, but most commonly from 5 to 10 MHz). The resonance frequency is determined by the properties of the bulk material, but it is also dependent on environmental conditions. Designs of high-quality quartz crystal resonators aim to minimize the effect of environmental perturbations [25]. For example, the orientation of the crystal cut is chosen to reduce the influence of temperature on the oscillation frequency. The resulting resonator has a high quality factor Q (10^3 to 10^6 or even more). If it is used in a positive feedback circuit, it generates a timing signal featuring an excellent short-term stability (in particular over observation intervals smaller than one second). The resonator is made of a plate of quartz crystal. Owing to the piezoelectric effect, mechanical deformations propagate in the bulk coupled with a variable electrical field [21]. The crystal enters resonance when the waves fulfill given boundary conditions. The quartz lattice is inexpensive and reliable, and it is the most common quartz used to build oscillators.

4.2.2.2 Voltage-Controlled Crystal Oscillators

A VCO is an oscillator whose output frequency is a Voltage- controlled, within given limits, by an external driving voltage. This device is mostly known as a Controlled Crystal Oscillator (VCXO). The VCXO is a quartz-crystal oscillator with a variable capacitance, somewhere along the positive-feedback loop, making possible the adjustment of the output frequency. This capacitor is often made of two parts: one fixed and the other varactor. The latter is regulated for oscillation-frequency fine tuning by an external DC voltage source [27].

4.2.2.3 Temperature-Compensated Crystal Oscillators

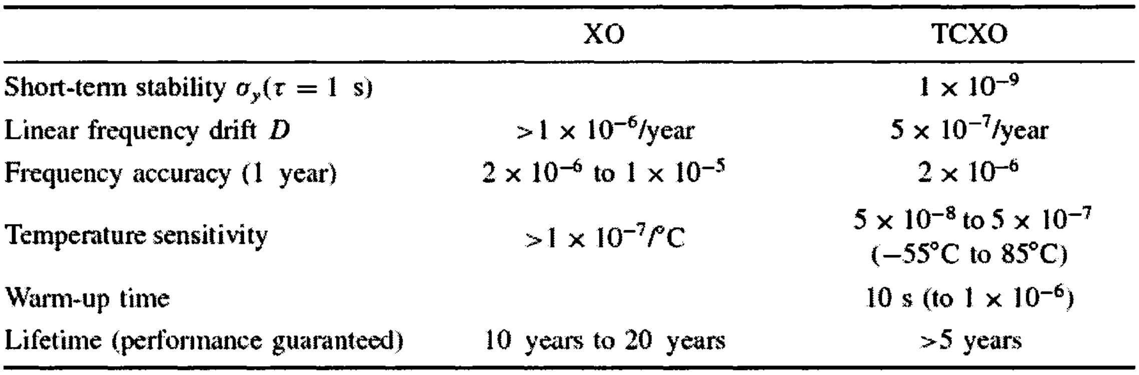
The plain XO's main problem is the dependence of its natural frequency on ageing, around 10^{-7}/day in plain models and on the temperature (in the order of 10^{-7}/C or above). Temperature-Compensated Crystal Oscillators (TCXOs) implement an automatic control on the oscillation frequency based on the measurement of the crystal temperature. This allows a frequency stability of 10^{-7} over a temperature interval from 0C to 50C. More sophisticated digital control models achieve frequency stability even in the order of 10^{-8} in the wider temperature interval from 0C to 70C [2].

4.2.2.4 Oven-Controlled Crystal Oscillators

Insulating the oscillator thermically is a much better method for working in a constant-temperature closed environment than using compensating temperature variations with a feedback control. Insulated clocks are called Oven-Controlled Crystal Oscillators (OCXOs). In OCXOs, the resonator and the other temperature-sensitive elements are placed in a controlled oven whose temperature is set at a point where the resonator frequency does not depend on temperature to minimize the effect of residual temperature variations. This achieves frequency stability values exceeding 10^{-9}/day. Double-oven temperature controls and auxiliary-plates crystal excitation achieve higher frequency stability in the order of 10^{-11}/day, which approaches the performance of some atomic secondary frequency standards (rubidium clocks) [28].

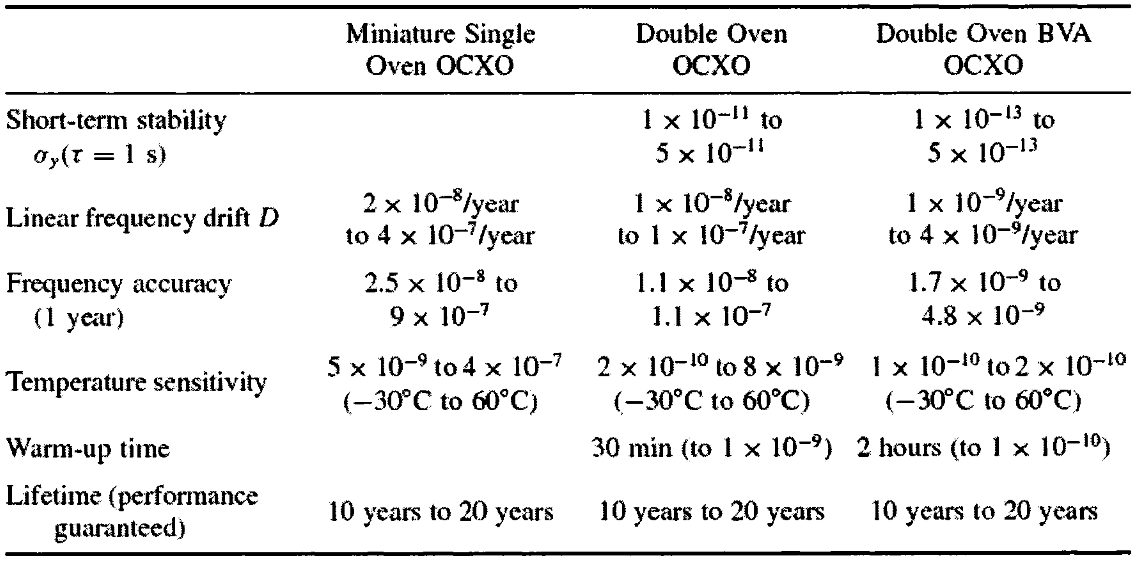
4.2.3 Performance and Characteristics of Crystal Oscillators

The typical performance and characteristics data of quartz oscillators available on the market are summarized in Table 4-1 and Table 4-2 below:



	XO	TCXO
Short-term stability $\sigma_y(\tau = 1\ \text{s})$		1×10^{-9}
Linear frequency drift D	$>1 \times 10^{-6}$/year	5×10^{-7}/year
Frequency accuracy (1 year)	2×10^{-6} to 1×10^{-5}	2×10^{-6}
Temperature sensitivity	$>1 \times 10^{-7}/°\text{C}$	5×10^{-8} to 5×10^{-7} (−55°C to 85°C)
Warm-up time		10 s (to 1×10^{-6})
Lifetime (performance guaranteed)	10 years to 20 years	>5 years

Table 4-1 – Typical Performance of Simple Quartz Oscillors – From (Bregini [2], p.286)



	Miniature Single Oven OCXO	Double Oven OCXO	Double Oven BVA OCXO
Short-term stability $\sigma_y(\tau = 1\ \text{s})$		1×10^{-11} to 5×10^{-11}	1×10^{-13} to 5×10^{-13}
Linear frequency drift D	2×10^{-8}/year to 4×10^{-7}/year	1×10^{-8}/year to 1×10^{-7}/year	1×10^{-9}/year to 4×10^{-9}/year
Frequency accuracy (1 year)	2.5×10^{-8} to 9×10^{-7}	1.1×10^{-8} to 1.1×10^{-7}	1.7×10^{-9} to 4.8×10^{-9}
Temperature sensitivity	5×10^{-9} to 4×10^{-7} (−30°C to 60°C)	2×10^{-10} to 8×10^{-9} (−30°C to 60°C)	1×10^{-10} to 2×10^{-10} (−30°C to 60°C)
Warm-up time		30 min (to 1×10^{-9})	2 hours (to 1×10^{-10})
Lifetime (performance guaranteed)	10 years to 20 years	10 years to 20 years	10 years to 20 years

Table 4-2 – Performance of High-Quality Quartz Oscillators – From (Bregini [2], p.287)

Clock performance is expressed in terms of the short-term frequency stability over one second (i.e., the Allan variance for τ = 1 s), of the linear frequency drift D, of the frequency accuracy expected over one year (vmax/vn) and of the temperature sensitivity (fractional frequency variation per Celsius degree or over a temperature range). Warm-up time is expressed as the time needed to achieve the frequency accuracy specified between brackets [2].

4.2.4 Atomic Frequency Standards

Atomic frequency standards, unlike quartz oscillators, are based on the intrinsic properties of atoms. Fundamental constants such as the energy gap between two quantum levels of certain atoms determine the reference frequency [1]. Currently, researchers argue that these atomic properties are the same everywhere in the universe, in space and in time, within known relativistic effects. Atomic clocks use the oscillation of an electromagnetic signal associated with a quantum transition between two energy levels in certain atoms as a frequency reference. The quantum bundle of electromagnetic energy is called a photon and is equal to the difference in energy between these two levels in one atom. Practical atomic frequency standards are currently based on exploiting the properties of three elements: 1) hydrogen (in the hydrogen-MASER frequency standard); 2) rubidium (in the rubidium-gascell frequency standard) and; 3) caesium (in the caesium-beam frequency standard). The principles of these three operating standards are outlined in the following section.

4.2.4.1 Physical Principle of Operation

Quantum Physics asserts that an atom can only be at quantum, discrete energy levels. Let E1 and E2 be the energy of two quantum levels 1 and 2, with E2 > E1. Then, the atom can transit between these two levels by absorption or emission of a photon of electromagnetic radiation at frequency ν_0 [2]. The value of the frequency ν_0 is determined by the energy conservation principle, according to the

Bohr's law

$E2 - E1 = h\nu_0$,

where $h = 6.625 \times 10{-}34\ J \cdot s$ is the Planck's constant.

The fact that the atom exhibits opposite magnetic moments whether it is in one or the other energy level is the basis of the first method. Hence, atoms from the two populations can be separated by deflection through an inhomogeneous magnetic field. This method is used in caesium-beam and hydrogen-MASER frequency standards. The second method is based on optical pumping, in an analogous fashion to what is done in

Erbium-Doped Fiber optical Amplifiers (EDFA). For example, an optical radiation of appropriate wavelength is used to excite (pump) atoms from level E2 to another level at higher energy E3. Spontaneous decay occurs quickly from levels E3 down to levels E1 and E2 and the net result is that the population of both level E1 and E2 increase. This principle is applied in rubidium and optically pumped caesium-beam frequency standards.

In atomic frequency standards the principle of operation is based on locking the resonance frequency v_0 by maximizing the number of atomic transitions between the two quantum energy levels 1 and 2. A feedback system is used to control the frequency that is synthesized from a quartz VCXO and that is used to probe the atomic resonator. In passive atomic frequency standards (e.g., the caesium-beam frequency standard), to excite transitions the microwave signal is applied to the atoms. By adjusting the frequency synthesized from the VCXO some feedback systems aim at maximizing the number of atomic transitions. Resonance is achieved when the frequency of the exciting microwave signal is v_0. In active atomic frequency standards, a device (e.g., the hydrogen-MASER active cavity) generates a self-sustained oscillation. An electronic system then phase-locks a VCXO to the oscillation generated. The principle of heterodyne detection must be used to lock the microwave frequency v_0 since the VCXO operates in a frequency range in the order of 1 to 100 MHz [11].

4.2.4.2 Caesium-Beam Frequency Standard

The caesium-beam frequency standard is based on the transition of atoms of the isotope 133 of caesium (133Cs) in the ground state between the hyperfine levels characterized by magnetic moment F = 4 (level 1, with energy E1) and F = 3 (level 2, with energy E2). An oven with a few grams of the isotope 133 of caesium (133Cs) effuses a beam of caesium atoms, distributed uniformly among 16 quantum energy levels, into a vacuum chamber. Then, the non-homogeneous magnetic field in the magnet 1 (polarizer) deflects the level-2 atoms through the resonant cavity (Ramsey resonator), where a microwave field irradiates them. The microwave signal has the precise frequency of v_0 = 9.192631770 GHz and stimulates atom transition to level 1 when the condition is fulfilled. Then, the non-homogeneous magnetic field in the magnet 2 (analyser) deflects only the level-1 atoms to a hot-wire ionization detector. The output current is proportional to the incoming 133Cs atom flux and to the transition probability from level 2 to level 1 [12].

A resonance peak is exhibited by the output current when the frequency of the probing microwave field is swept across the value ν_0. The spectral line width around ν_0 is normally in the range of 100 Hz, due to the flight time of atoms through the cavity. Finally, by the detector the signal output is used to steer a quartz VCXO, from which the microwave radiation and the output timing signal are synthesized, which aims to maximize the number of transitions. Therefore, the short-term stability of the quartz oscillator is coupled with the long-term stability of the steering atomic resonator. The value of ν_0 of unperturbed 133Cs atoms has been used to define the second in the International System (SI), as 9 192 631 770 times the resonance oscillation period $1/\nu_0$.

4.2.4.3 Hydrogen-MASER Frequency Standard

The principle of operation of a hydrogen-MASER (Microwave Amplification by Stimulated Emission of Radiation) frequency standard is based on the stimulated emission of electromagnetic radiation at the frequency ν_0. This corresponds to the transition of hydrogen atoms between the states having magnetic moment F = 1 (level 2) and F = 0 (level 1). First, a beam of hydrogen atoms is selected by the polarizer magnet, which produces a magnetic field of intensity 1 T. Next, atoms at the upper energy level E2 (F = 1) are injected into a storage bulb. They are surrounded by a high-Q microwave resonant cavity tuned to the ν_0 = 1.42040575177 GHz atomic resonance frequency. To shield the interaction region from the ambient magnetic field the resonant cavity is exposed to an internal homogeneous magnetic field of intensity 10^{-7} T [28].

In the bulb, atoms bounce around and decay to the lower energy level E1 (F = 0), by emitting radiation at the frequency ν_0. The atoms produce a microwave field (active MASER) that stimulates this emission. An external probing field (passive MASER) may also be used. The coating of the inner surface is thought to have the highest number of atoms elastic collisions as possible. The average interaction time achieved with the microwave field is in the order of 1 s. Finally, hydrogen atoms are continuously pumped away from the cavity, to maintain a constant pressure not higher than 10^{-5} Pa.

In active hydrogen MASER, the cavity volume is big enough to self-sustain oscillation. Therefore, the microwave radiation at frequency v_0 emitted by the cavity is detected by an antenna and used to lock a quartz VCXO, which generates the output timing signal, by way of frequency multiplication and mixing (heterodyne detection). The radiation power detectable by the antenna outside the cavity is minute; between −100 dBm and −110 dBm (i.e., 100 fW to 10 fW). The short and medium-term stability of active hydrogen MASERs is thought to be achievable in practical clocks. Their long-term stability is affected by mechanical shocks and temperature variations because the resonance frequency of the cavity depends on its physical dimensions. Therefore, very effective mechanical and thermal shields must surround the cavity. The result being that active hydrogen MASER clocks are large and expensive.

In passive hydrogen MASER, the cavity volume is smaller. A microwave-probing field must be applied to stimulate emission, which is not self-sustained. The servomechanism to control the output frequency is similar to that used in caesium-beam clocks. A passive MASER is, in general, less sensitive to environmental variations than active MASERs, but it has less short-term stability [1].

4.2.4.4 Rubidium-Gas-Cell Frequency Standard

The rubidium-gas-cell frequency standard is based on the transition of atoms of the isotope 87 of rubidium (87Rb) between the base levels characterized by magnetic moments F = 1 (level E1) and F = 2 (level E2). Atom state selection and transition detection are achieved by optical pumping. Light from a lamp filled with 87Rb is filtered through a cell (hyperfine filter) containing 85Rb vapor, before it excites 87Rb atoms in a cell (absorption cell) filled with buffer gas inside a resonant microwave cavity. The purpose of the buffer gas (a mixture of inert gases) is to scheme of principle of a rubidium-gas-cell frequency standard increase the interaction time of the 87Rb atoms with the microwave field in the resonator, by causing lots of elastic collisions of the 87Rb atoms with the buffer atoms before an inelastic collision with the cell walls.

The hyperfine filter cell allows only the light spectrum component νA, which optically pumps atoms at the lower level E1 (F = 1), to reach the absorption cell. Here, this level is depopulated as 87Rb atoms absorb the light radiation, and are pumped to the upper level E3 and then decay down to both the base levels E1 and E2. If the level E1 is depopulated, the cell then becomes transparent to radiation νA. Nevertheless, to excite their transition to level E1 again microwave probing radiation at frequency νp =6.3846826128 GHz is applied to the atoms. Hence, this level is repopulated and optical absorption starts over [15].

4.2.4.5 Performance of Atomic Frequency Standards

Clock performance for quartz oscillators is expressed in terms of the short-term frequency stability over τ seconds (i.e., the Allan variance), of the linear frequency drift D (expressed in parts ν/νn per year), of the frequency accuracy expected over one year (νmax/νn) and of the temperature sensitivity (fractional frequency variation per Celsius degree or over a temperature range). The warm-up time is also expressed as the time needed to achieve the frequency accuracy specified between brackets [14]. Some typical performance data and characteristics of atomic frequency standards are summarized in Table 4-3:

	Rubidium standard	Caesium-beam standard	Hydrogen MASER standard
Short-term stability $\sigma_y(\tau)$	for $\tau = 1$ s: 5×10^{-12} to 5×10^{-11}	for $\tau = 100$ s: $<10^{-13}$ (prim. std.)[a] to 3×10^{-12}	for $\tau = 1$ s: 2×10^{-13} (active) 2×10^{-12} (passive)
Linear frequency drift D	5×10^{-11}/year to 5×10^{-10}/year	0	$<10^{-13}$/year to 5×10^{-12}/year
Frequency accuracy (1 year)	1×10^{-10} to 1×10^{-9}[b]	1×10^{-14} (prim. std.) to 7×10^{-12}	10^{-12}
Temperature sensitivity	1×10^{-12}/°C to 1×10^{-11}/°C	1×10^{-14}/°C to 1×10^{-13}/°C	1×10^{-14}/°C
Sensitivity to magnetic fields (per 10^{-4} T)	5×10^{-12} to 2×10^{-11}	$<1 \times 10^{-14}$ to 1×10^{-12}	3×10^{-14} (active) 1×10^{-14} (passive)
Resonance frequency ν_0	6.3846826128 GHz	9.192631770 GHz[c]	1.42040575177 GHz
Warm-up time	2 min to 30 min (to 5×10^{-10})	30 min (to 3×10^{-12})	24 hours (to 1×10^{-12})
Lifetime (performance guaranteed)	up to 10 years	3 years to 10 years	3 years
Basic wearout mechanism	Rb lamp and cavity life (15 years)	Cs beam tube (3 to 10 years)	Ion pumps and H_2 source depletion, cavity resonance shift (7 years)
Portability and intended location	space, air, ground	ground and laboratory oriented	Definitely ground and laboratory oriented
Typical volume	$<10^{-3}$ m^3 to 10^{-2} m^3	$<10^{-2}$ m^3	0.5 m^3 (active)
Typical weight	0.5 kg to 2 kg	10 kg to 30 kg	>50 kg
Power consumption	10 W to 50 W	25 W to 50 W	70 W to 100 W

Table 4-3 – Performance Characteristics of Atomic Clocks – From (Bregini [2], p.292)

4.2.5 The Global Positioning System

The Global Positioning Systems (GPS) is not a "clock", but a complex system of clocks and satellites. GPS receivers are often used as additional references for Stand-Alone Synchronization Equipment (SASE) clocks, especially in wide-area synchronization networks to compensate for the issue of timing transfers over long distances. GPS is a satellite radio system that provides continuous real time dimensional positions, velocity and time information to equipped users anywhere on Earth. Originally designed for navigation and positioning, it is also used to disseminate precise times, time intervals and frequencies. GPS receivers are also used to time-synchronize Base Transceiver Stations (BTSs) in mobile telephone cellular networks based on CDMA [2][30].

4.2.5.1 How Does GPS Work?

GPS systems are based on a set of 24 orbiting satellites (space segment), each carrying two atomic clocks on board and they are monitored by ground control stations (control segment). Each satellite broadcasts spread-spectrum signals bearing unique pseudo-random codes with very long periods. Five to eight satellites can be seen from Earth at any given time and the principle of triangulation is used to determine the receiver's position. Precise distance measurements (range measurements) to four satellites yield a set of four equations for four unknowns: latitude, longitude, altitude and GPS-system time [2].

The distance of the GPS receiver from each observable satellite is found by measuring the time of flight of the signal received from that satellite. This measurement is found by determining the relative phase error between the received pseudo-random code and one generated locally and the measurement from one extra satellite removes the ambiguity of the local time alignment. The receiver's computer can estimate accurately its distance from each satellite in view as the signal speed is known, this speed is the speed of light in vacuum with some additional minor delays due to the crossing the neutral and ionized parts of the atmosphere. The triangulation of range measurements is based on the assumption that the exact location of the satellites is known [29]. The highest precision is used to determine GPS-satellite orbits and the receivers have stored memory almanacs that list the position of each satellite listed moment by moment. Minor variations in satellite orbits are measured by the GPS control stations to compute correction data that are uploaded to satellites once or twice a day.

4.2.5.2 Use of GPS for Time and Frequency Dissemination

GPS can also be used as a pure time reference to disseminate precise times (GPS system time), time intervals and frequencies. It is available worldwide, with no charge, to any user equipped with the equipment. For this reason, GPS receivers are often used as additional references at the input of SASE clocks in synchronization networks, or to time-synchronize Base Transceiver Stations in mobile telephone cellular networks based on CDMA.

To set precise time, an independent source determines the three-dimensional position of the receiver to exploit the full potential of GPS. A position error of 1 m yields about 3 ns of time error and about 1 MHz of frequency error, due to the variation in the signal propagation delay. Moreover, Space Vehicle (SV) time at the receiver must be corrected for errors of the SV clock with respect to the GPS time and for periodic relativistic effects. The SV clock error is transmitted in each data frame as a set of polynomial coefficients, based on uploaded Master Core Switch control data, while the relativistic correction is computed from the SV orbital parameters normally used for SV position determination [32][33].

At present time or time intervals can be produced by the receiver and receivers at different locations can be time-synchronized to GPS time. If synchronization to UTC is required, the UTC correction in the Navigation Message can be applied. The accuracy of GPS time signals is related to the ability of the receiver to track the received Coarse/Acquisition (C/A) code. Accuracies in the order of 100 ns are possible with undegraded GPS signals (i.e., with no SA) and correct receiver positions. A GPS receiver can also be used as a pure frequency source. Frequency synchronization can be established by steering a local oscillator using integrated code-phase measurements or by directly measuring the SV carrier frequency. In the latter case, carrier frequency tracking is mostly accomplished by phase locking to the de-spread carrier. The transmitted carrier signals are controlled by the on-board atomic clocks and are very stable and accurate. The Doppler shift that results from SV orbital moving must be corrected properly. Frequency accuracy in the order of 10^{-12} is possible with undegraded signals (i.e., with no SA).

The common-mode time transfer is a useful technique for improving the accuracy of GPS time synchronization and it is analogous to Differential GPS (DGPS). If two receivers track a satellite at the same time, it can be assumed that the path-delay corrections, the SV-time error and the data-message error are the same for both received signals. Therefore, by comparing one receiver to a reference clock, such as a national Precise Positioning Service (PRC), the error in the SV time measured at that reference location can be transmitted to the other location and then used to adjust the clock there. Tests over paths of thousands of kilometers showed time transfer accuracy within 25 ns [31].

4.2.6 Performance in AAPN

Most of the clocks described can be used in AAPN and clocks are essential for synchronization. In the solution chapter, a pattern will be explained for synchronization and how to accomplish it. The comparison tables show that atomic clocks are very stable in addition to GPS clocks. A disadvantage with GPS is that its accuracy decreases in thunderstorms and other weather disruptions since GPS uses its satellites to reset and configure its clocks. Aside from this, GPS is the easiest to configure and use since it provides absolute time and the clocks are already synchronized via satellites. Its accuracy is also very high. If AAPN is applied in a stable weather environment with few weather disturbances then clocks locked to GPS clocks are the best option.

Quartz clocks (crystal clocks) need more time and more frequent maintenance messages in the initial phase of synchronization. The VCXO and the Double Oven clocks are the most applicable since their performance is better than normal crystal clocks and they are relatively inexpensive. Costs are important since numerous clocks are required in this situation. One for the core, which is the primary clock and it should be the most stable, and a clock per core node for every selector and edge node that they are connected to. Atomic clocks are the best clocks in AAPN for synchronization, but they are expensive. The cesium standard clocks are extremely stable and they remain efficient for longer periods of time but they can not be used in AAPN because of the cost. A better choice is the rubidium standard because their accuracy is high and its short-term drifting is also very low, and they are less expensive.

4.2.7 Summary

The operation principle of clocks consists of a generator of oscillations and an automatic counter of such oscillations. A frequency source supplies a (pseudo) periodic signal whose meaningful information is frequency, which is usable as a reference for frequency synchronization. A time source, supplies a signal that carries absolute (e.g., UTC) or relative time information. A primary frequency standard is a source of standard frequency that is usable as a signal reference for frequency synchronization, which does not need to be steered by any other external reference. On the other hand, a secondary frequency standard is a source of standard frequency that can be, and usually is, steered by a primary standard source.

Quartz oscillators are based on the piezoelectric effect: a mechanical strain in the crystal yields an electrical field and vice versa. An XO is an electronic oscillator, where a quartz crystal is excited by a periodic electrical signal at the resonance frequency. To stir up and maintain oscillation the crystal resonator is inserted into a positive feedback loop. The VCXO is a quartz-crystal oscillator, where a variable capacity allows for the fine adjustment of the output frequency. Based on the measurement of the crystal temperature TCXOs implement an automatic control on the oscillation frequency. In OCXOs, the resonator and the other temperature-sensitive elements are placed in a controlled oven, where the temperature is set to a point where the resonator frequency does not depend on temperature.

An atomic clock uses the oscillation of an electromagnetic signal associated with a quantum transition between two energy levels in an atom. In practical atomic clocks, the energy levels are determined by the hyperfine interaction and are characterized by different magnetic moments. Atomic frequency standards are based on locking the resonance frequency by maximizing the number of atomic transitions between the two quantum energy levels. A feedback system is used to control the frequency used to probe the atomic resonator.

The caesium-beam frequency standard is based on the transition of 133Cs atoms between two hyperfine levels. Hydrogen-MASER frequency standards are based on the stimulated emission of electromagnetic radiation at the frequency that corresponds to the transition of H atoms between two hyperfine levels. The rubidium-gas-cell frequency standard is based on the transition of 87Rb atoms between two hyperfine levels and on optical pumping to achieve atom state selection and transition detection [2].

GPS is a satellite system that provides continuous three-dimensional position in real time, velocity and time information to people with the proper equipment anywhere on Earth. GPS is based on a set of 24 orbiting satellites (space segment), each carrying two atomic clocks on board and is continuously monitored by ground control stations (control segment) that compute precise orbital data and clock correction parameters for each satellite. The triangulation principle is used to determine the receiver's position, which is based on range measurements from a minimum of four satellites. GPS can be used also as a pure time reference to disseminate precise time, time intervals and frequencies with excellent accuracy. With undegraded GPS signals, time accuracy in the order of 100 ns and frequency accuracy in the order of 10^{-12} are possible. With SA, the SPS allows predictable time accuracy in the order of 340 ns (for 95% of the time) and frequency control accuracy in the order of 10^{-12}. Careful use of common-mode techniques can substantially reduce SA's impact on time transfer, when needed.

4.3 Network Time Protocol

NTP is one of the application services protocols within the TCP/IP suite. NTP finds itself in the company of the TCP/IP workhorses, which includes Telnet, FTP, SNMP, SMTP, and more. NTP has been assigned the port number 123, and NTP implementations rely on UDP for transport services. The following information is derived from references [7][9]. Despite NTP's membership in the dominant TCP/IP suite, the protocol can be adapted to and incorporated into other protocol stacks. The fundamental purpose of NTP is simple: time synchronization among participating network devices with reference to a reliable clock source. From the network administration perspective, a reliable NTP clock source could be either absolute or relative [8].

The effect of NTP operations on a network is straightforward: accurate time synchronization among the configured devices. The NTP algorithms, however, are complex since an internet network may be composed of numerous routers, switches, and servers from different vendors running different versions of operating systems, even from the same vendor. Thus, NTP implementations must be interoperable across this spectrum.

- Multiple WAN links of varying capacities may introduce varying levels of delays into the network traffic, including the NTP messages. The NTP algorithm must take into account the very nature of the network on which it operates, including the potential for use of asymmetric paths between the communicating devices.
- If network security cannot be assured through other means, NTP itself must be able to detect the false ticker and also ensure secure time synchronization exchanges in a scalable manner.

All of the preceding issues and more must be considered in the design and updating of the NTP. NTP has a well-structured algorithm but it works under the assumption that propagation delay is half the round trip time, which is similar to the first proposed solution [10]. Therefore, by that assumption it is mentioned in the protocol that if wanting to synchronize terminals that are widely separated by distance, hence great propagation delay, then a time server should also be allocated at the other end.

4.4 Available Solutions

This section will state different available methods. These solutions vary depending on application performance and availability. Some methods described in references such as [34] and [35] may actually apply as solutions to the problem. However, the proposed methods are better since they address the problem directly with its parameters.

4.4.1 NTP

NTP is a network time protocol as explained explicitly in section 1.1. Moreover [34] discusses the solution of synchronization using NTP protocol. Therefore it can be shown the differences between [34] and the available methods.

In [34] using the NTP protocol the time server must have a clock which uses absolute time like GPS clocks. This is like the method 1 which is proposed in 0. All these methods can not determine or even start the synchronization process without the availability of absolute time source. Because the propagation delay is not known and can not be calculated except with absolute time. NTP and [34] take the assumption that the upstream and downstream are equivalent and they are half the round trip time minus the processing times. Meanwhile these methods will fail if this assumption is not true and the propagation delays are very different. In this case the clocks will be out of synchronization and the only item that can get them back into calculations if they have an absolute time source like GPS. Meanwhile Method 2 proposes a solution independent of propagation delays.

4.4.2 Distributed Slot Synchronization

The Distributed Slot Synchronization method is the closest method to the available Method 1, as they almost do the same things. However there are some differences which will be discussed later on.

A brief description of the Distributed Slot Synchronization method is taking from [35].

"Distributed slot synchronization (DSS) is a network-wide packet synchronization technique which coordinates node transmissions so that packets arrive aligned to one another at a reference point in the network, independent of propagation delays. DSS was developed for use in the contention resolution with delay-lines (CORD) project, a DARPA-funded 2.5 Gbit/s/λ, wavelength division multiplexer (WDM) optical packet-switched network test bed. In this implementation, it is experimentally demonstrated that the DSS system, operating with 80 MHz control logic, achieves a packet arrival jitter of less than 13 ns with 12 km node spacing. DSS was also shown to be robust against noise and node failure or fiber breaks. The technique is data rate and format independent and can be used in other star, extended ring, or tree-and-branch network architectures for metropolitan area network (MAN) and access applications." [35]

This method is propagation delay independent as it does not include propagation delays into its calculations or synchronization technique. Moreover the synchronization technique is similar to Method 2 discussed in 5.2.2 but there are some differences which are as follows. Firstly, in DSS one of the nodes is designated the master node to which the other slave nodes align themselves and if the master node fails then one of the slave nodes takes over as master. Meanwhile in Method 2 as it is applied for AAPN, the master nodes are the core nodes assigned from the beginning. DSS works for 50 nodes maximum while AAPN needs to work with 1024 nodes. DSS therefore fails to meet the AAPN requirements. Finally the separation of distance for DSS to work properly should be less than 12 km, while in AAPN it can go up to 1000 km. Other than these differences the algorithm is almost the same.

There is a proposed solution mentioned in patent [3], but there is no any performance analysis on how this algorithm works. Therefore it can not be applied in this thesis.

5 Synchronization Method

This chapter will describe one solution in detail. A description of the process will be given prior to the detailed explanation. A State Diagram of the system is given in (Figure 5-1) showing three possible states of synchronization. It's assumed that the system consists of a selector connected to an input port at the Core Node.

The Idle State is where the Core Node can detect no activity on the input port. The lack of activity may be because there is no Edge Node connected to that input port, or that there has been a failure causing loss of signal. If an intelligible signal is detected at the input port there is a transition to the Initialization State. In this state there is a protocol to align the clocks to the required tolerance. Once both the Edge and Core Nodes are satisfied with the alignment the Edge Node can initiate the transfer of data to other Edge Nodes on the network, and the synchronization protocol enters the Maintenance State.

There are two processes executed in the Maintenance State: Alignment Maintenance and Synchronization Detection. Alignment Maintenance keeps the clock at the Edge Node within tolerance in the face of clock drift and minor (slow) changes of propagation delay. Synchronization Detection involves a keep-alive check to ensure that the system is aligned sufficiently to maintain the transfer of user data. A major change in the propagation delay (such as might be caused by a protection switch) will cause a loss of synchronization to be detected. The clocks have to be realigned to the new propagation delay, and the system returns to the Initialization State. In the Initialization State if the clocks fail to align the System assumes communication has failed, and transitions to the Idle State.

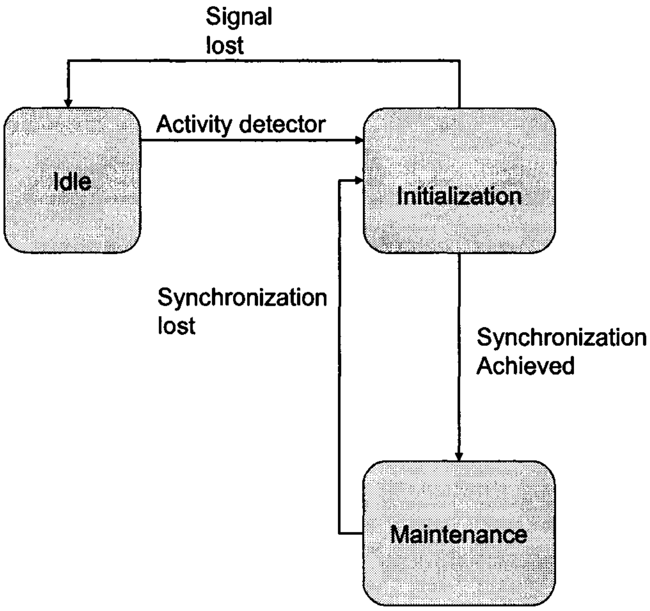


Figure 5-1 – State Diagram for Synchronization Scheme

When a Selector is included between the Core Node and the Edge Node the synchronization must be performed in two stages. First, the Selector must be aligned with the Core Node, and then the Edge Nodes must be aligned to the Selector. The following sections outline the implementation of this diagram, and compare the performance of the different methods.

5.1 Synchronization Components

The Core Node, Selector and Edge Node are part of the synchronization scheme and all require hardware for synchronization purposes. Where a Selector is used it is first synchronized and aligned to the Core Node, then the connected Edge Nodes synchronize and align to the Selector. The description of the process will be based on a Selector-Core Node process.

A master clock is located at the Core Node. This clock is divided to identify timeslot boundaries, and further divided to provide a timeslot sequence number. The Timeslot

Sequence Number must be sufficiently long to avoid ambiguity with long round-trip times.

Selectors and Edge Nodes have slave clocks. These clocks are free-running prior to synchronization. They are also divided to identify timeslot boundaries and timeslot sequence numbers. In the initialization phase they are required to generate timestamps to a level of resolution sufficient to achieve the required tolerance on alignment. These clocks differ from the Master clock in that they will use an oscillator that will be capable of being closely matched in frequency to the master (either phase locked or a high-quality clock), and the counters will be capable of being reset as part of the alignment process.

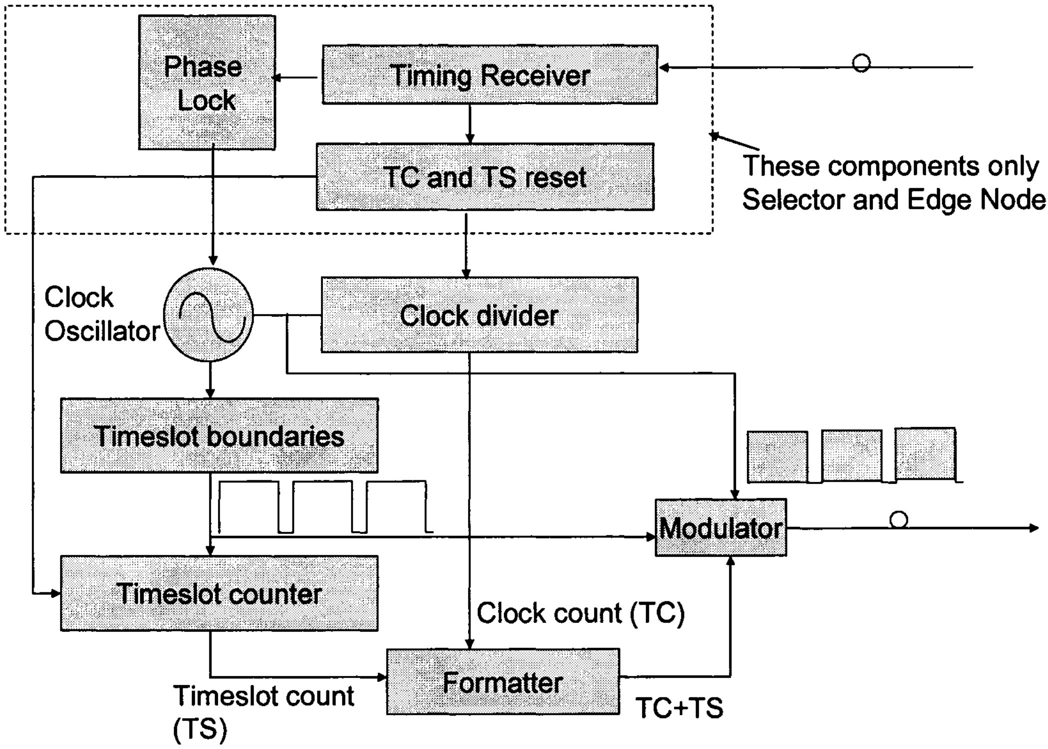


Figure 5-2 – Timing Circuits in Edge Node, Selector and Core Node

The timing circuits in the Edge Node, Selector and Core Node are essentially the same (Figure 5-2). The Clock Oscillators are initially free-running. Their frequencies must be equal to or an integer multiple of the transmission line rate. Their outputs are fed into two dividers: the first generates the format of the timeslots and guard periods; the second is a binary division that gives a clock count (TC). The output of the timeslot counter is further divided to provide a timeslot count (TS). The length of the timeslot count must be sufficient to not repeat a given value within a period greater than the round-trip time for a worst case distance to the Core Node. The Edge Node and Selector have additional circuitry over the Core Node to permit the phase of their clocks to be locked to the Core Node clock. Within a given timeslot the value of TS will remain constant.

During synchronization the Formatter combines the Clock and Timeslot counts (TC and TS) and feeds the current value to the modulator. The modulator takes the current values and modulates them on to the optical carrier using the local Clock. The values of TC/TS are delineated by flags. The data in each timeslot is preceded by a preamble that allows the receiving data clock to phase-lock to the incoming data. A communication channel provides a bidirectional capability for passing messages for the purpose of establishing and maintaining the alignment. An example of the communication paths has been given in Figure 3-8, and the architecture in this diagram will be used to describe the process.

5.2 Activity Detection

When a Selector is first activated the Core Node does not know of its existence. There is an input port on the Core Switch that is connected to the Selector, and the Core Switch has to be able to detect that the Selector has been activated. This is achieved by the Selector sending intelligible photonic signals to the Core Node. The Core Node can only detect these signals if it occasionally makes a connection through the switch to sample each idle input port.

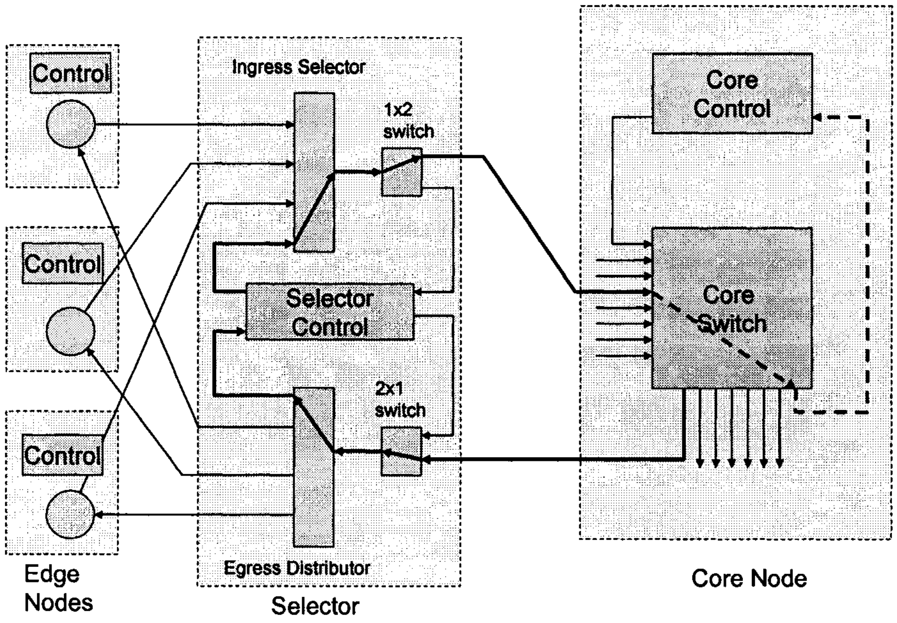


Figure 5-3 - Initial configuration of Control Channels

Referring to Figure 5-3 the Ingress Selector and Egress Distributor and associated switches will be switched continuously to the connections to and from the Core Switch. Assuming that the Core Switch is already passing user data between ports that have already been initialized so a continuous connection to and from the Core Node Controller to the idle input ports is not possible and initially there is no path for a signal to traverse the Core Switch and reach the Core Node controller. To allow a new device to be detected the Core Node must schedule an occasional test timeslot to sense activity on ports which have not been initialized. This sensing is indicated by a dotted line through the Core Switch.

On its initial start-up the Selector will continuously send dummy timeslots to the Core Switch even if the port is closed (Figure 5-4). These dummy timeslots are to inform the Core Switch that there is activity on that link. When the dummy timeslots arrive at a time when the port is being sensed, the signal from the new device will be detected by the Core Controller.

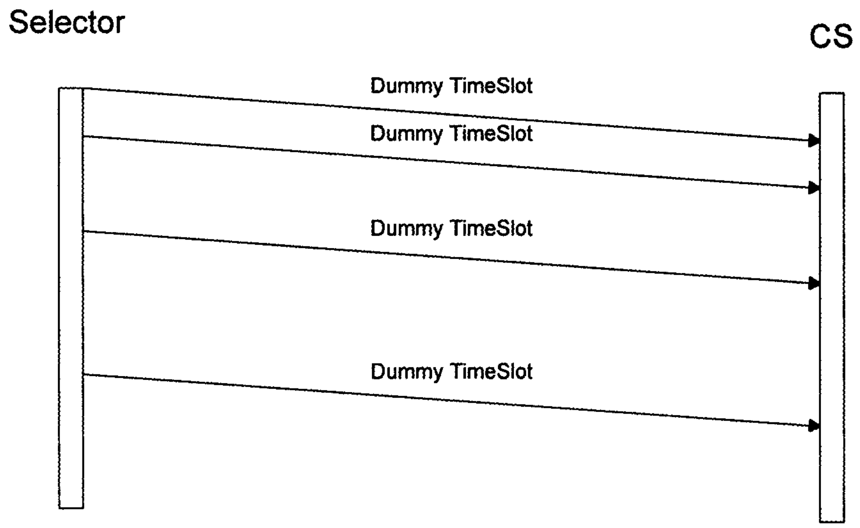


Figure 5-4 - Dummy Timeslots

The format of a dummy timeslot during activity detection is given as follows:

Preamble Flag TSa TC1 Flag TSa TC2 Flag TSa TC3 Flag …

The dummy timeslots appear on the input port of the Core Switch (Figure 5-5). However, at this stage the timeslot boundaries are not aligned with the Master Clock. The result will be that when the switch is opened for sensing, two partial timeslots will be passed. The first fragment to arrive will generally not have a preamble and will be discarded. The second fragment will have a preamble, and the TS and TC counts in the payload can then be decoded.

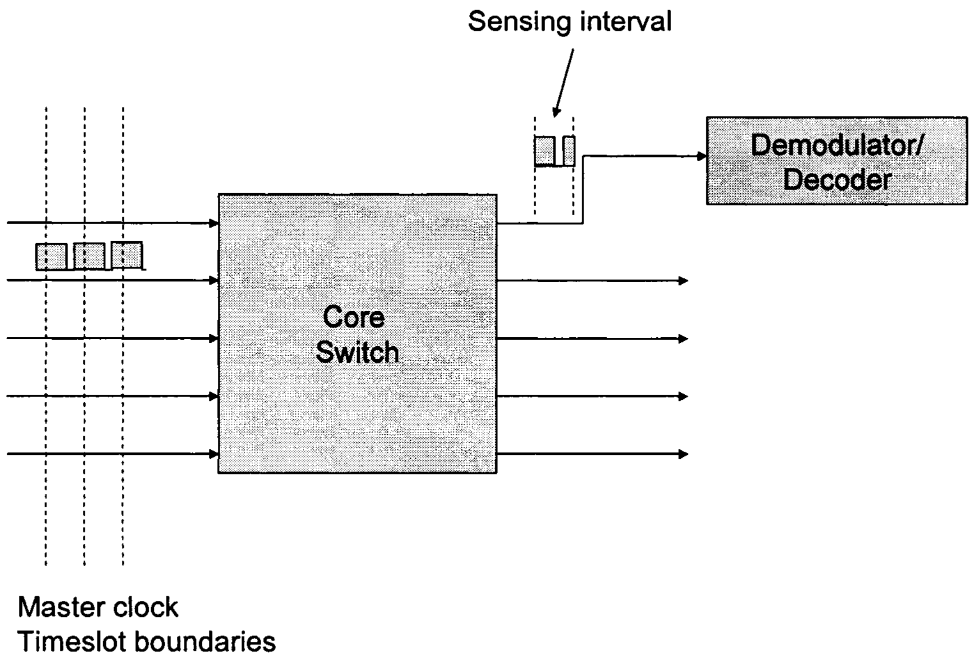


Figure 5-5 – Block Diagram of Core Node During Activity Detection

Once the activity is detected the Core Node will move to the Initialization State. Two basic methods of Initialization are described below, followed by the use of GPS clocks for their enhancement.

5.2.1 Method 1 – Round-Trip Time Measurement

With the first method the Core Node measures the Round-Trip Time (RTT). The Core Node subtracts RTT from its current value of TS and TC, and transmits the result to the Edge Node/Selector. The Edge Node/Selector can reset its Clock and Timeslot counters to the received value, and the result will be an approximate alignment. Following this approximate alignment an averaging process is used to merge the clock alignment to within the required tolerance.

When activity has been detected the Core Node has to schedule the measurement of RTT. The Core Node selects a convenient timeslot, and sends a preamble and timestamp

(TS+TC) to the Selector. When the preamble is received by the Selector it will switch the output of the optical receiver directly back the transmitter. The received Timestamp will be returned directly to the Core Node (Figure 5-6). To receive the timeslot the Core Node has to hold a switch connection closed from the input port to its controller until the returned timestamp is received. The RTT can be calculated by subtracting the values in the received timeslot from the current values of the Core Node counters. It is assumed that the propagation delay in each direction of transmission is approximately equal, and the required time value at the Edge Node/Selector is half the RTT in advance of the Core Node clock. This process may be repeated to obtain an average value.

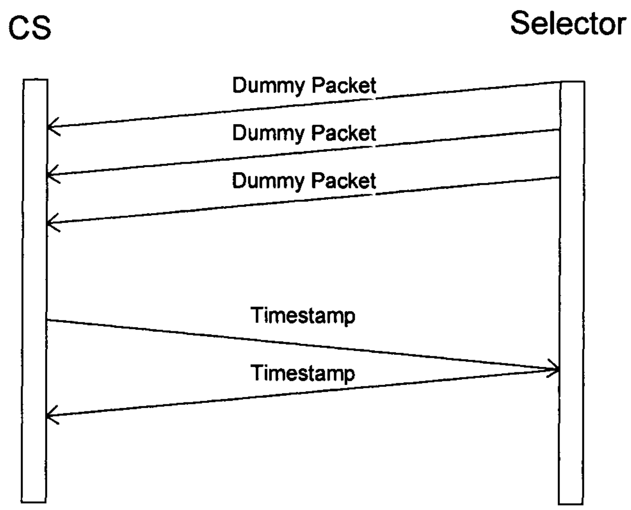


Figure 5-6 – Core Switch sends Timestamps

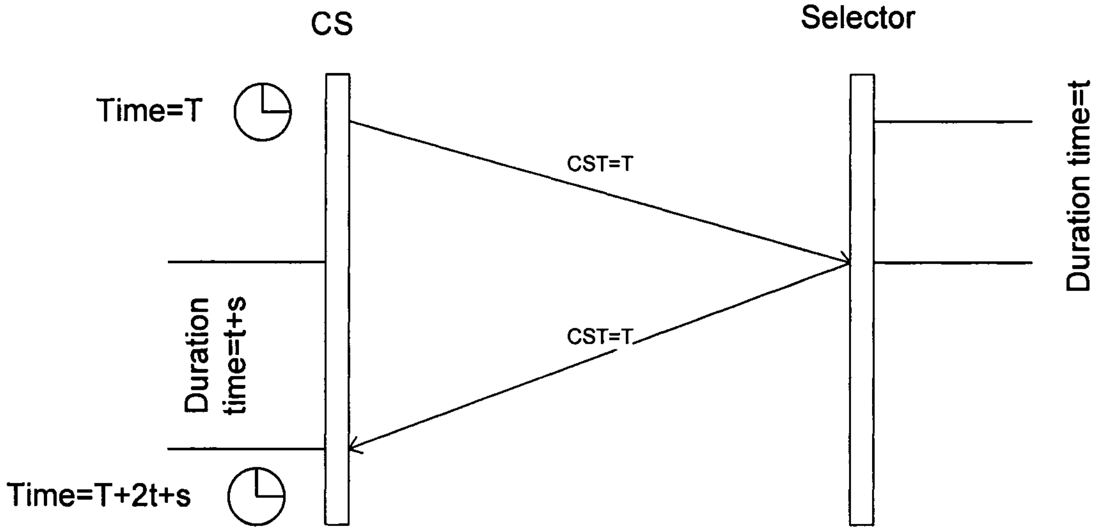


Figure 5-7 – Message Flows for Calculating Round-Trip Time

The Core Switch sends the timestamp at time=T. CST is the Core Switch Time. Propagation delay to the selector will take time=t. The propagation delay for the instant replay will be time= t+s. The case where the propagation delays are different will be discussed later on in this chapter. While *s* is a very small positive or negative value which represents the difference in propagation delay due to temperature and other factors. The Core Switch then calculates the one-way propagation delay by dividing the roundtrip time (RTT) by 2. Therefore the time take for the packet to go from Core Switch to selector is t (Figure 5-7).

After calculating the propagation delay, the Core Switch sends a timeslot to the selector informing it to change its clock. The content of the message is the current value of the Core Node clock minus RTT. When the timeslot reaches the selector the clock times for both the Core Switch and selector will be approximately the same (Figure 5-8). Also in this message is an instruction for the Edge Node/Selector to send a message back to the Core Node in a specified timeslot, this time depending on the free timeslot the Core Switch has allocated for synchronization.

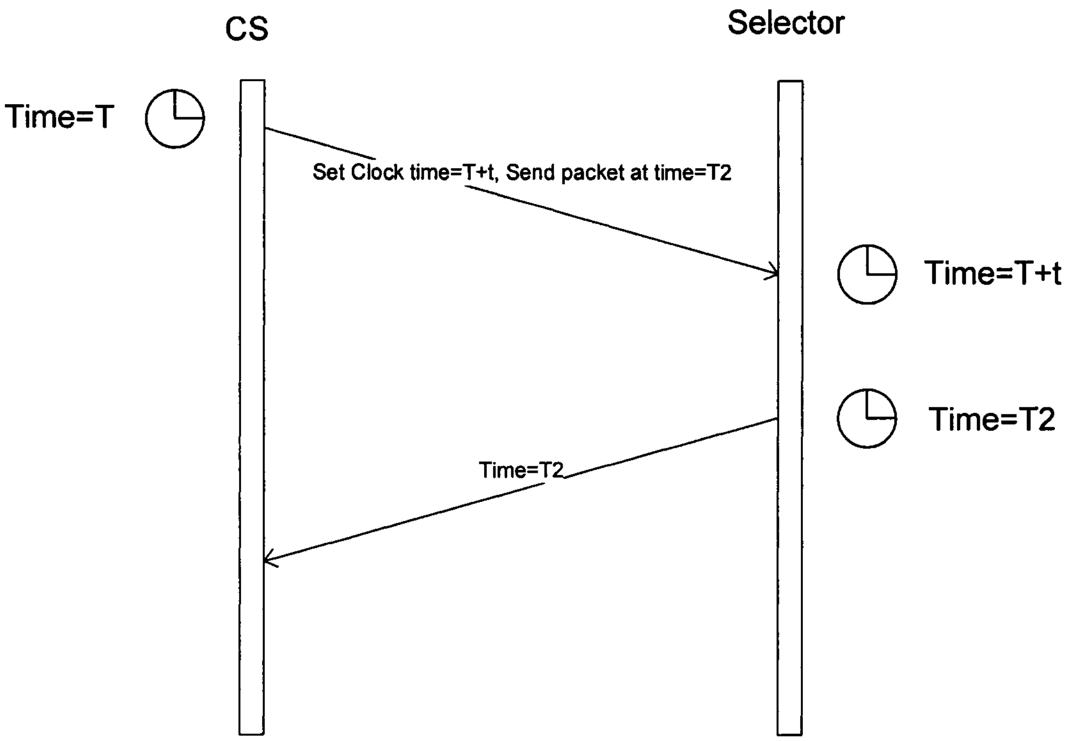


Figure 5-8 - Selector Sets Requested Time

The time Core Switch asks the selector to send a pulse message to indicate its estimated position when it reaches the middle of the timeslot. As the clocks are not yet fully synchronized there will generally be an error in its position when received at the Core Node.

Edge Switch2	Edge Switch7	Free	Sync. ↑	Edge Switch8	Edge Switch3	Sync	Free

T2+t

Table 5-1 – One Timeslot table in the Core Switch

Table 5-1 is a time slot diagram and it shows that T2+t is in the middle of the Sync timeslot. If the message sent by the selector arrives at exactly this time it means that the clocks are synchronized. If the pulse does not arrive at the centre of the timeslot, then the difference is calculated. The Core Switch then sends a correction message to the selector to reset its clock to adjust its clock by an amount equivalent to the error (Figure 5-9).

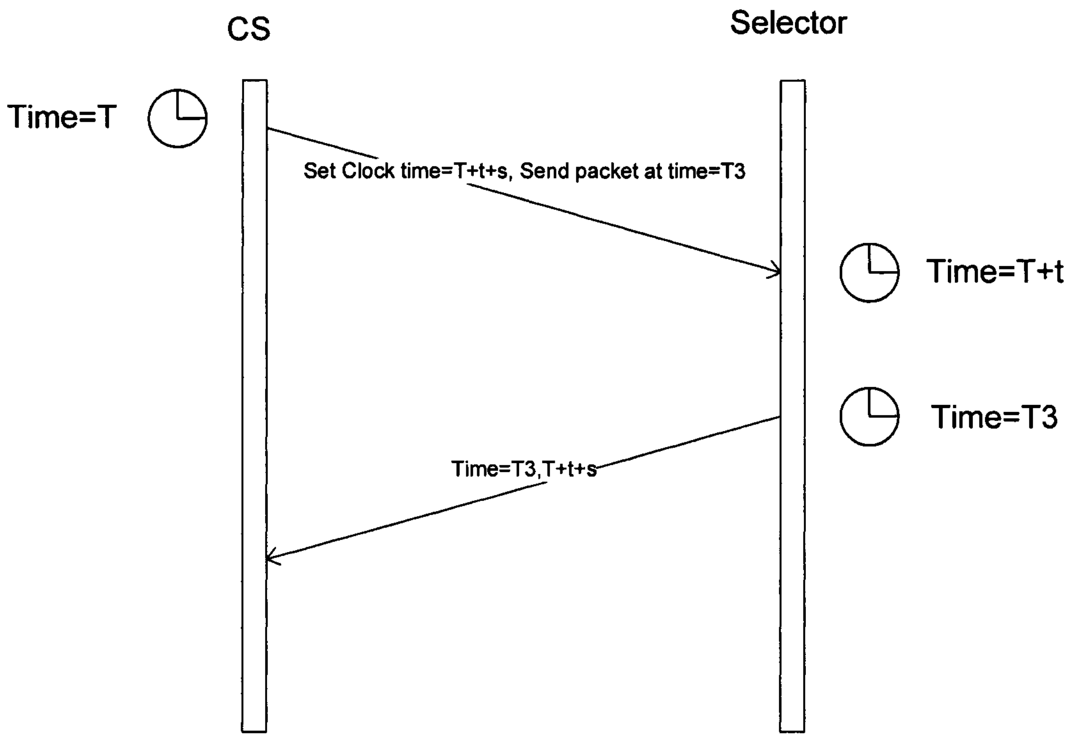


Figure 5-9 - Iterations for fixing errors

If the difference in the first trial is *s*, then the Core Node asks the selector to reset its time to a difference of *s*. This process is ongoing until the required accuracy is acquired from both clocks. The whole process is outline in Figure 5-10, the flowchart for the alignment process for the Core Node is (Figure 5-11) and the Edge Node is (Figure 5-12).

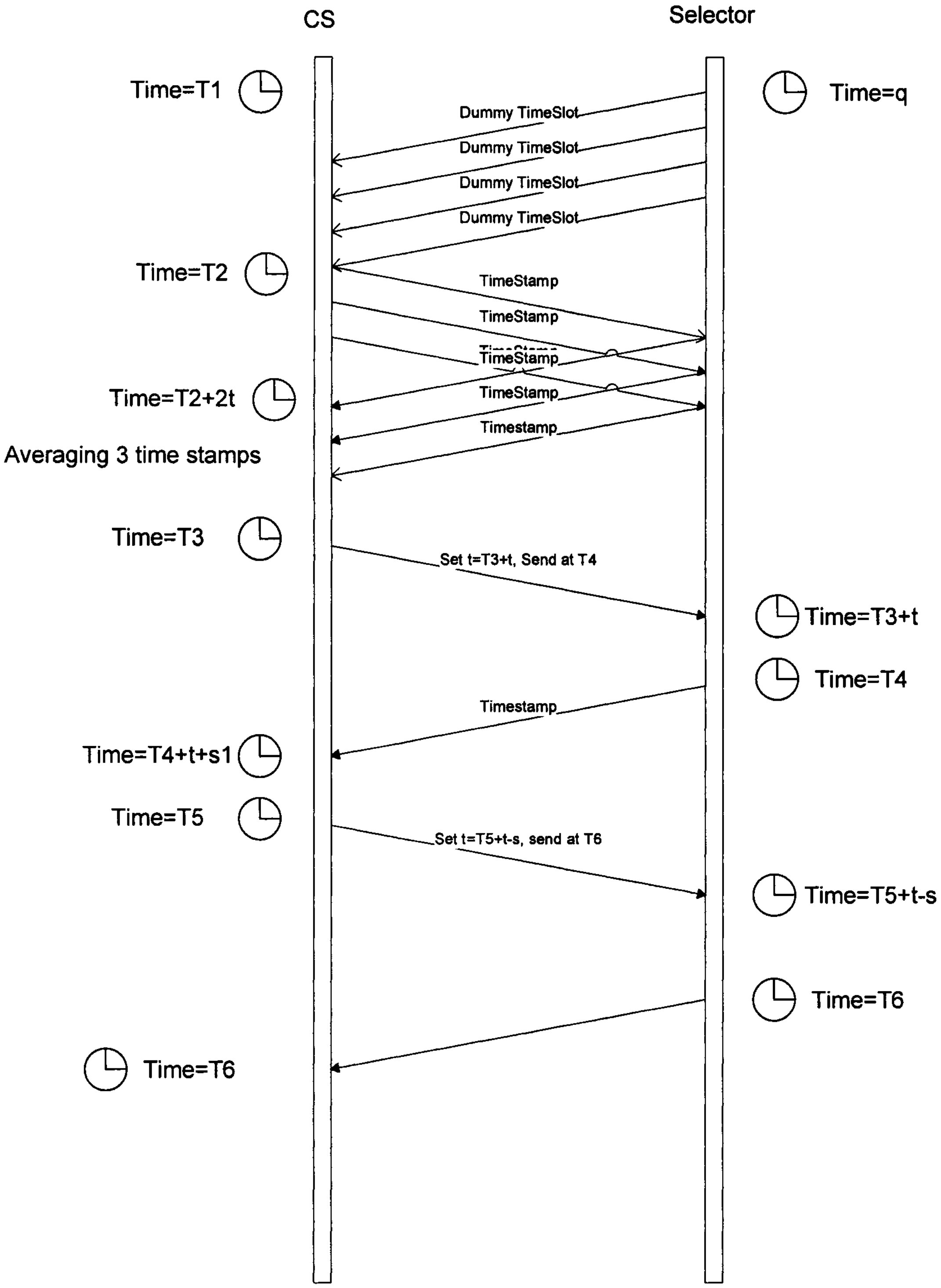


Figure 5-10 - Process for aligning clocks

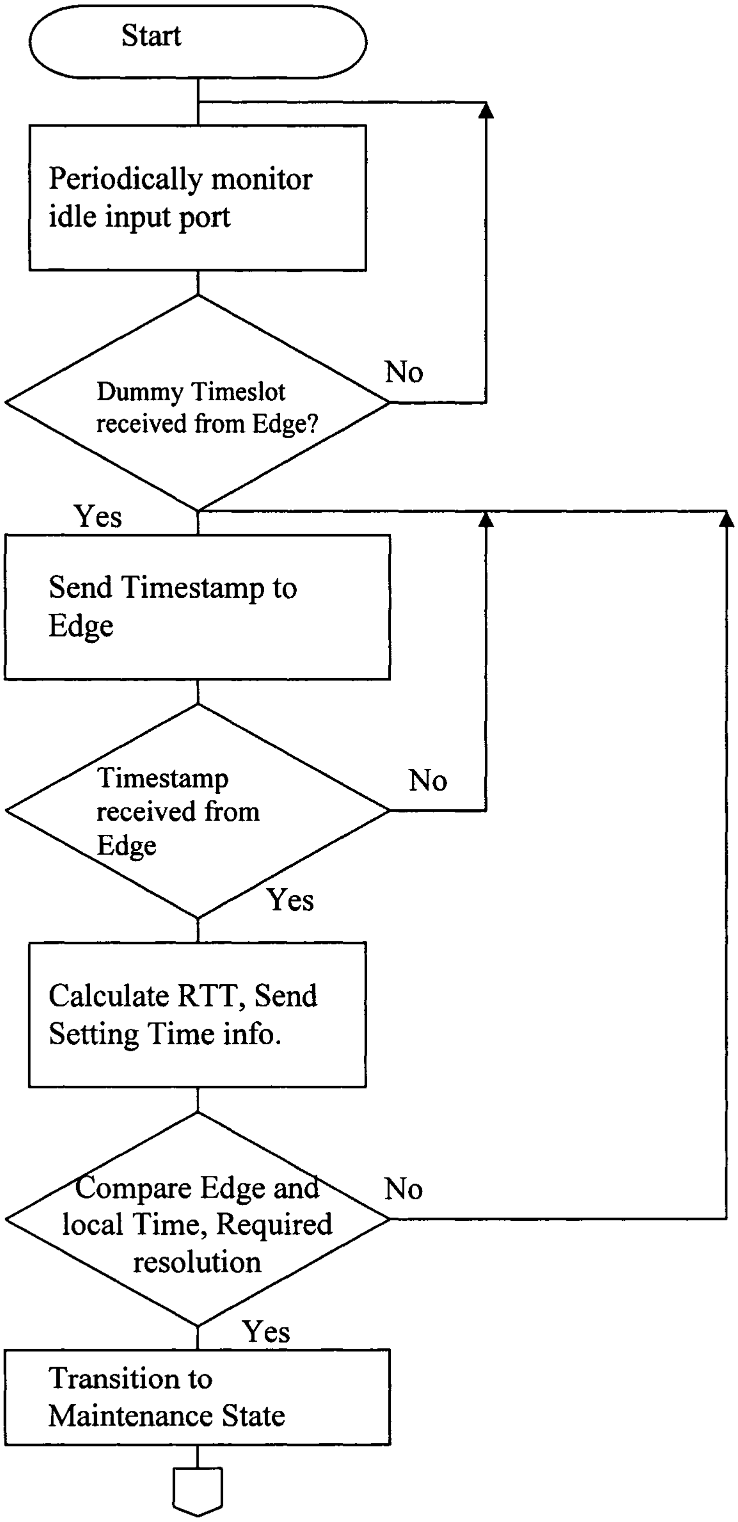

Start
Periodically monitor idle input port
Dummy Timeslot received from Edge?
No
Yes
Send Timestamp to Edge
Timestamp received from Edge
No
Yes
Calculate RTT, Send Setting Time info.
Compare Edge and local Time, Required resolution
No
Yes
Transition to Maintenance State

Figure 5-11 – Core Node Alignment Process

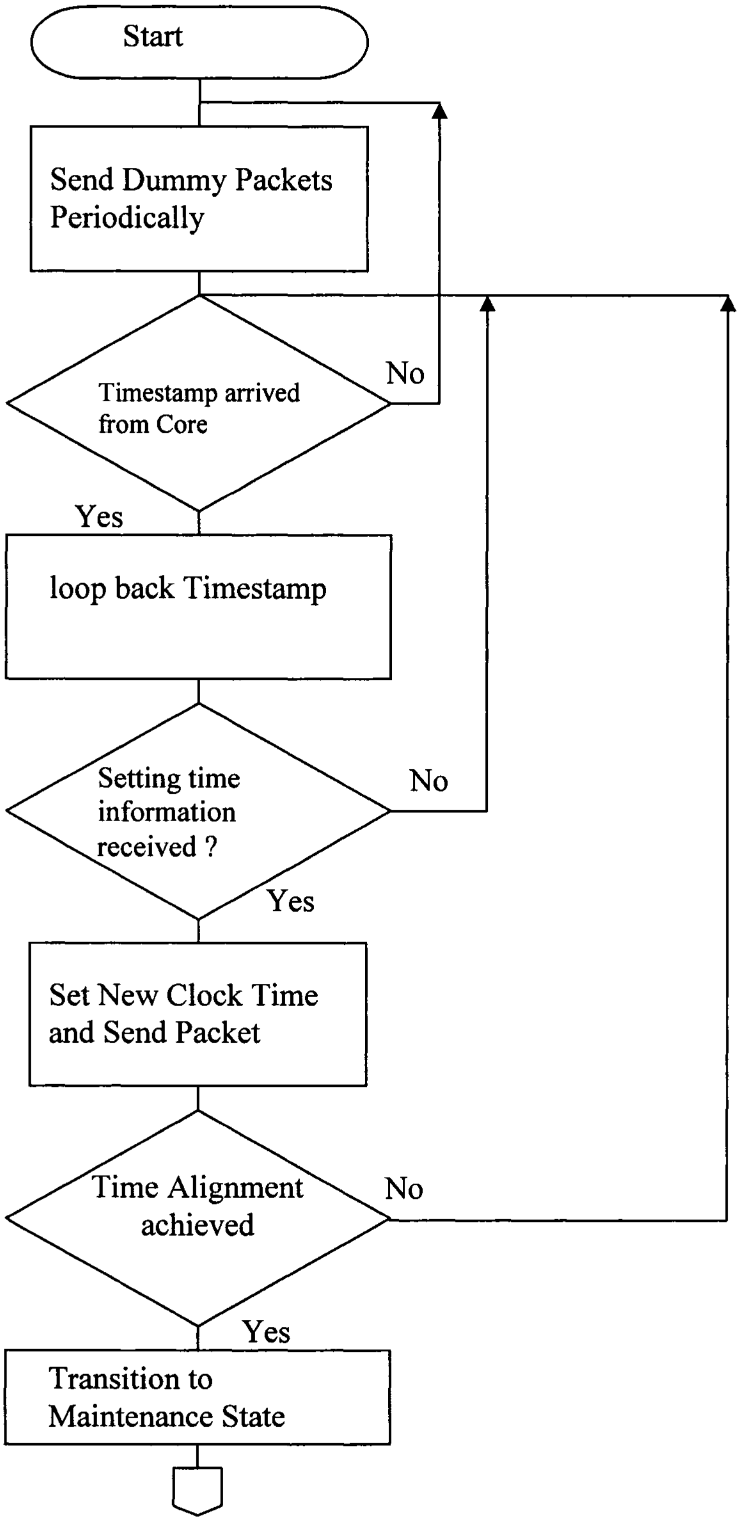

Start
Send Dummy Packets Periodically
Timestamp arrived from Core
No
Yes
loop back Timestamp
Setting time information received ?
No
Yes
Set New Clock Time and Send Packet
Time Alignment achieved
No
Yes
Transition to Maintenance State

Figure 5-12 – Edge Node Alignment Process

This method converges in all times and the number of times for synchronization depends on the accuracy needed. In AAPN there should be a maximum timing error of. This error will be discussed thoroughly in the performance chapter. Once the required accuracy has been achieved the system can enter the maintenance state.

5.2.2 Method 2 – Offset Timestamp Method

Once activity has been detected the Core Node has to schedule a timeslot for a return message. Generally, there is a delay of at least one timeslot before there is a transmission opportunity. The received timeslot count is included in the message along with the Core Node timeslot number. The received values of the counts TS and TC are compared to the current values indicated by the Master Clock. The difference between the values received and the Master clock-derived values gives the current offset of the Selector Clock from an ideal position. For example:

Clock rate	1GHz (time resolution 1nsec)
Timeslot width (including guard period)	10μsec
Timeslot counter	16bits (~650msec repetition cycle)

Combined with the above configuration the information sent by the Selector should be its timeslot number and the value of the free running counter. The Core Switch will receive this message and add to it, its current timeslot number and free running counter. An example to illustrate Edge switch is TS3 and TC=234(nsec). At the instant that the Core Node receives this message its values of TS and TC are 2 and 123. If the system were perfectly aligned these would also be the values received from the Selector. The Selector is therefore running one timeslot and 111nsec ahead. The Core Node can then send a correction message to the Selector (TS = -1, TC = -111) and on receipt the Edge Node will adjust its clock by that amount. However, during the Round-Trip-Time (RTT) the time count will drift slightly because the frequencies of the clocks at the Core and Edge Nodes are not exactly the same. The signal in the timeslot received from the core must be used to phase-lock the clock at the Selector to synchronize the frequencies.

The synchronization of frequencies is a basic phase lock loop problem. The signal is passed by a multiplier or a sequential-logic phase (such as an r-s flip flop) followed by a low pass filter. This basic design is shown in Figure 5-15. The dummy timeslots from the Edge Node now contain values of TS and TC that are close to those required. The process is repeated and as soon as the tolerance on the alignment is achieved, a message is sent to the Edge Node to move to its Maintenance State. The alignment process is illustrated by the flowcharts in Figure 5-13 (Edge Node/Selector) and Figure 5-14 (Core Node)

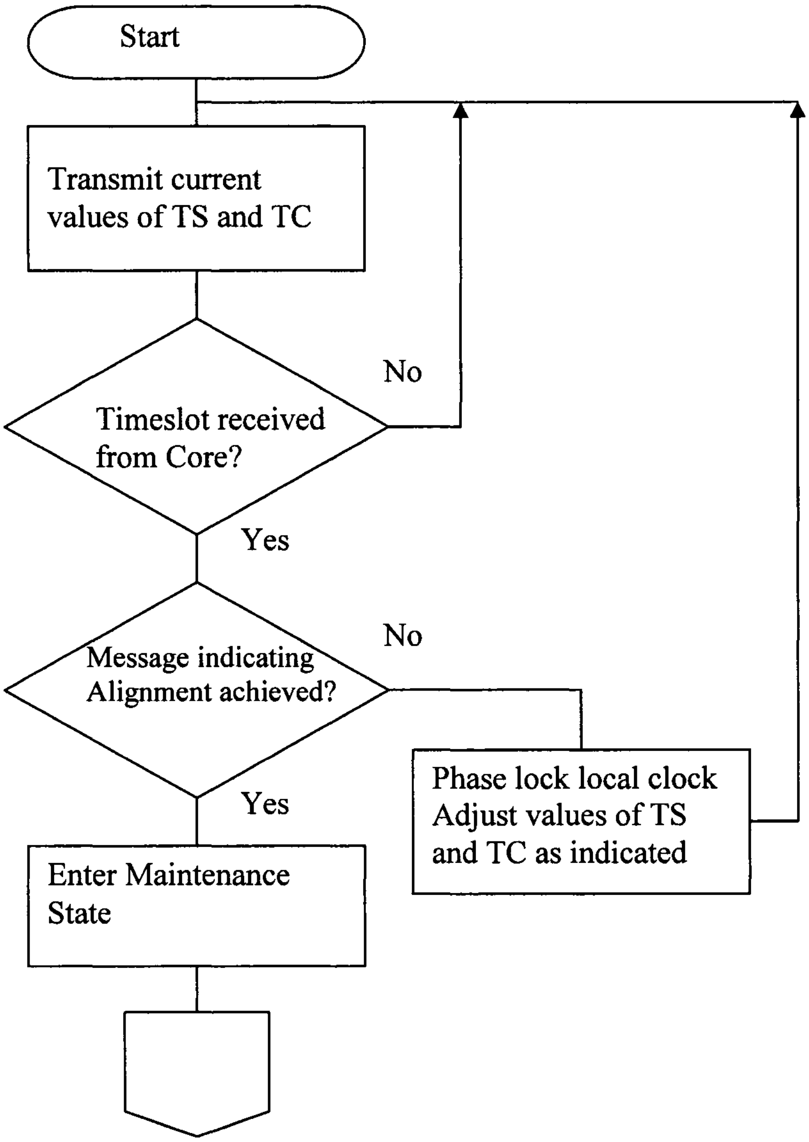

Start
Transmit current values of TS and TC
Timeslot received from Core?
No
Yes
Message indicating Alignment achieved?
No
Yes
Phase lock local clock Adjust values of TS and TC as indicated
Enter Maintenance State

Figure 5-13 - Edge Node Alignment Process

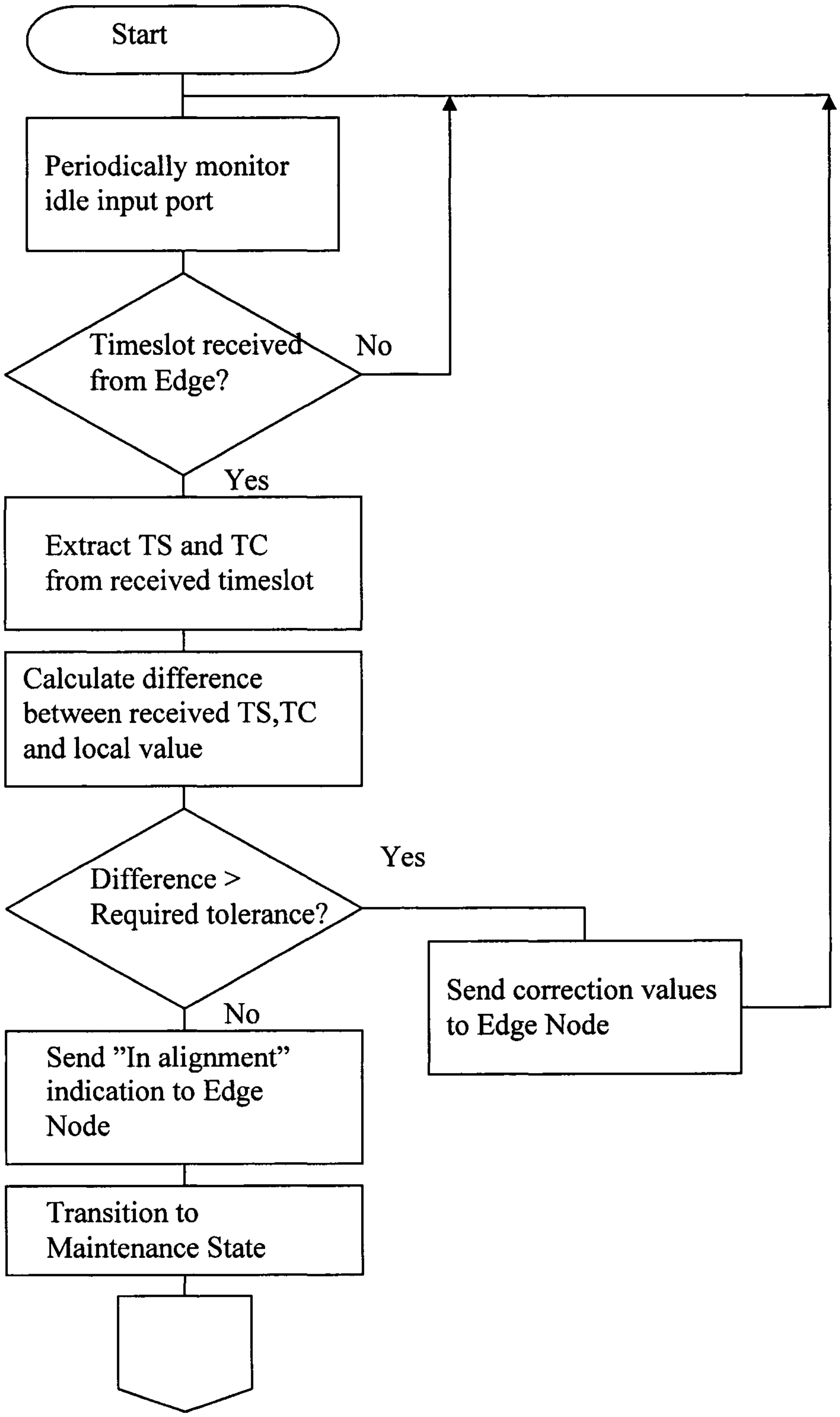

Start
Periodically monitor idle input port
Timeslot received from Edge?
No
Yes
Extract TS and TC from received timeslot
Calculate difference between received TS,TC and local value
Difference > Required tolerance?
Yes
Send correction values to Edge Node
No
Send ”In alignment” indication to Edge Node
Transition to Maintenance State

Figure 5-14 - Initialization - Core Node Process

Assuming that the timeslot counter and the free running counter for Core Switch and Edge Switch are T1 and T2 respectively to get the best results T1 and T2 must be exact. T1-T2 must =0, but in reality it is T1-T2=t and the value of t will be sent back to the Edge Switch. At this stage the Edge Switch will adjust its timing and timeslots to have proposed values. This process continues until t reaches the required resolution.

For best performance the Core Node clock and the Selector/Edge Node clocks have to be in the same phase. If they are not, then the drifting rate will be high and the initialization phase will take longer and iterations. This solution is a basic phase lock loop. A complete description of a phase-locked loop is given in [17][18][19][20]. The signal should be passed by a multiplier or a sequential-logic phase like r-s flip flop and then a low pass filter. The basic design is shown in Figure 5-15:

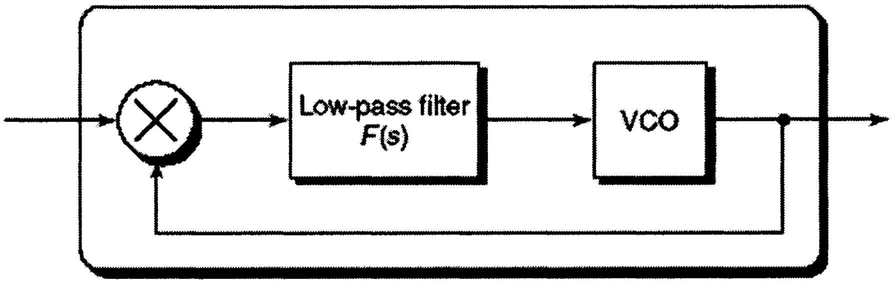


Figure 5-15 - Clock Phase/Frequency Alignment

5.2.3 Use of GPS Clocks

Both of the above methods can be simplified if the clocks at the Core Nodes, Edge Nodes and Selectors are referenced to GPS clocks. In this case the clocks are set to the same value. For Method 1 the values of TS and TC at the Edge Node are derived from the GPS clock, and the Timeslot received at the Core Node include the current values at the Edge Node/Selector. The difference between the local and received values at the Core Node directly gives the propagation delay from Edge Node/Selector to the Core Node. The return message from the Core Node to the Edge Node/Selector the return propagation delay can be calculated as well as the clock offset. In the second method the

Timestamp will reflect the Universal Time at the Core Node by checking the difference at the Edge Node/Selector the one-way propagation delay can also be calculated.

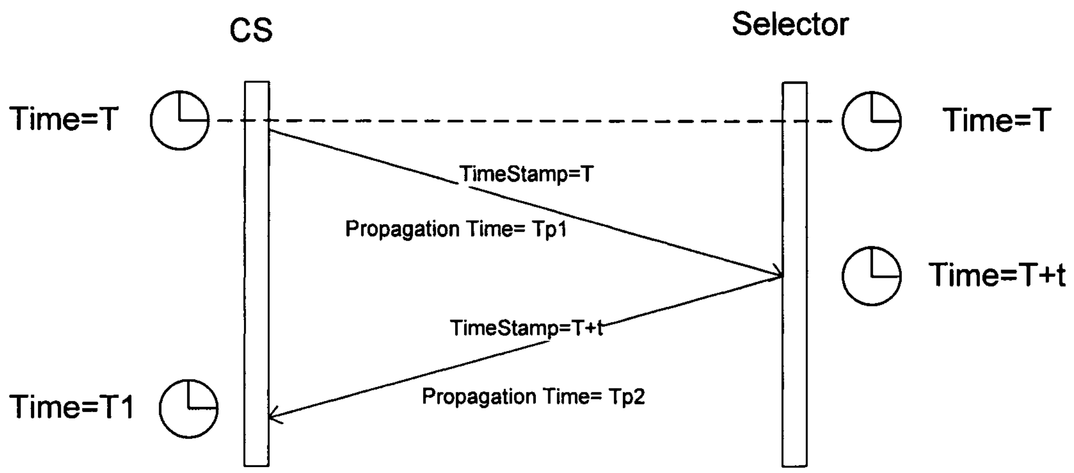


Figure 5-16 - RTT Using GPS clocks

As shown in (Figure 5-16) Tp1 and Tp2 can be calculated from the timestamp values set by the Core Switch and the Selector. At this stage of the Synchronization process the only task left to do is for the Core Switch to inform the Selector/Edge Switch the beginning of the timeslots, and when to send them.

5.3 Maintenance

The maintenance phase will depend on the type of the clocks used but is in general, the phase where handles with any clock drifting and variations in propagation delay due to temperature and other effects. Maintenance occurs after the 2 devices are fully initialized and synchronized. As the types of clocks and performance will be discussed in later chapters, the maintenance process will be summarized.

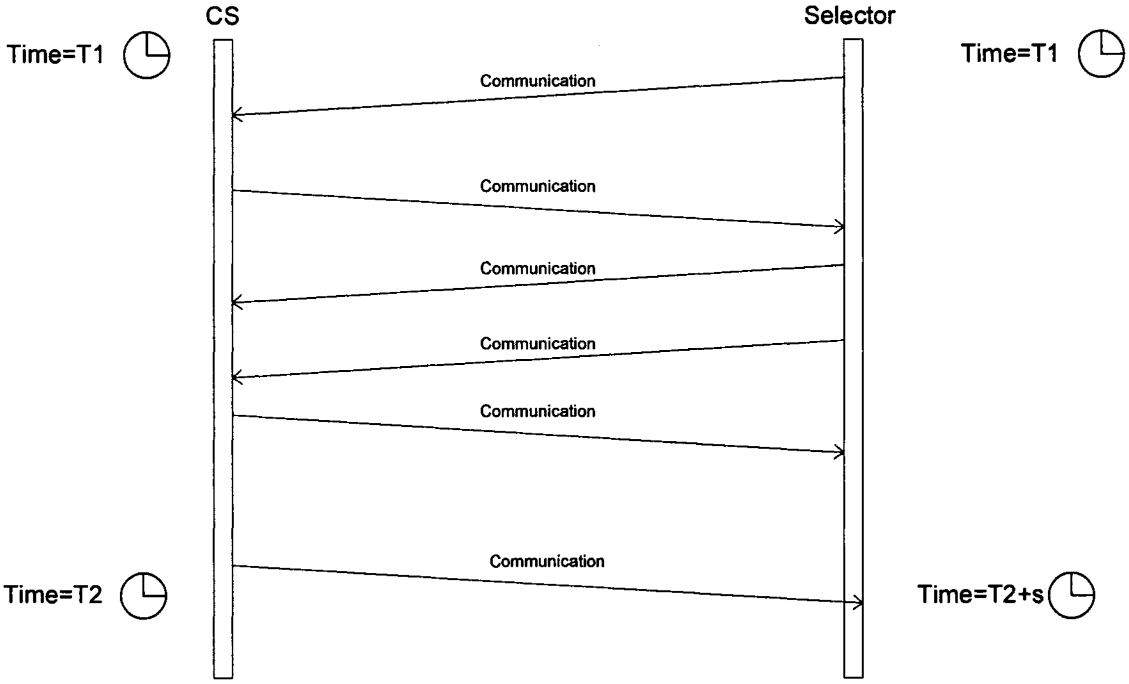


Figure 5-17 - Drifting of Clocks

The *s* in (Figure 5-17) is the small drift in time caused by the clock and the maintenance process should minimize *s*. To insure that the value of *s* is minimal and eliminated, time synchronization messages should be done between the devices at regular intervals. The core switch has control timeslots, one of these timeslots is for synchronization, and this includes initialization and maintenance. Therefore after initialization this slot will be used for maintenance. The following diagram (Figure 5-18) outlines the process of maintenance in detail:

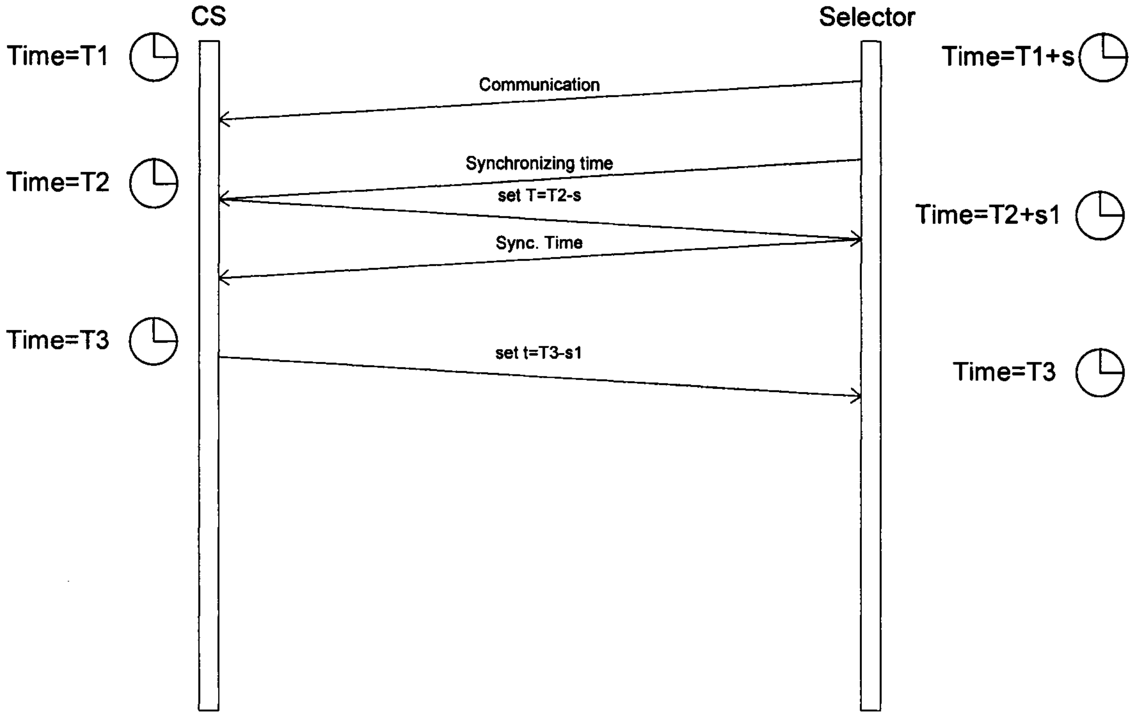


Figure 5-18 – Maintenance Process

As shown in Figure 5-18 the same process of synchronization from the initialization phase is applied here, but it should take less time and the change in times of the clocks is smaller. The most important part of the maintenance phase, are the update intervals and this is explained in the performance section.

The Maintenance State must be capable of detecting failures in the synchronization, and failures in the transmission path. Synchronization failures may be due to some changes in the propagation delays of the underlying photonic transmission. For example, a protection-switch to a path with a different propagation causes delays to the original path. The change in propagation delay will result in Timeslots being received at the Core Node in the wrong position in time and they will not be switched correctly. In this case, the maintenance messages from the Edge Node Selector to the Core Node will not get through. A timer with a value T_f at the Core Node will start each time a Synchronization Message is received. T_f is chosen to be longer than the repetition interval of the Synchronization Messages. If the timer expires it indicates a loss of synchronization or a loss of communication and the Core Node will send a “Synchronization Failure” message to the Edge Node/Selector and revert to the Initialization State.

The Edge Node/Selector requires a similar timer that is reset each time a clock update message is received. If this timer expires it indicates a failure in the transmission from Core Node to Edge Node/Selector. A “Transmission Failure” message is sent to the Core Node, and the Edge Node Selector enters the Initialization State. The Edge Node/Selector will also enter the Initialization State if it receives a ‘Synchronization Failure” message from the Core Node.

6 Performance Analysis

The performance of the clocks is specified in terms of the following parameters:

- Clock acquisition time

- Clock drift rates and "Keepalive" intervals

Actual implementation of the algorithms was not considered necessary or practical. The method of evaluation chosen was to write a simulation of the behavior in software.

6.1 Method 1

As mentioned above the timeslot should reach the Core Switch at an exact specific moment such that it will be synchronized as shown in (Figure 6-1).

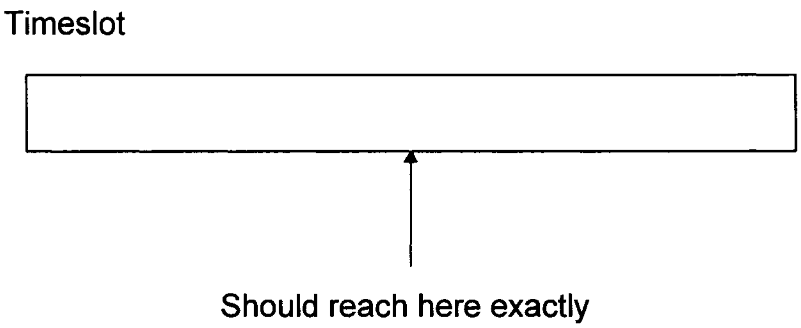


Figure 6-1 – Time needed to synchronize

To look in details of a full round trip we should get the following Figure 6-2.

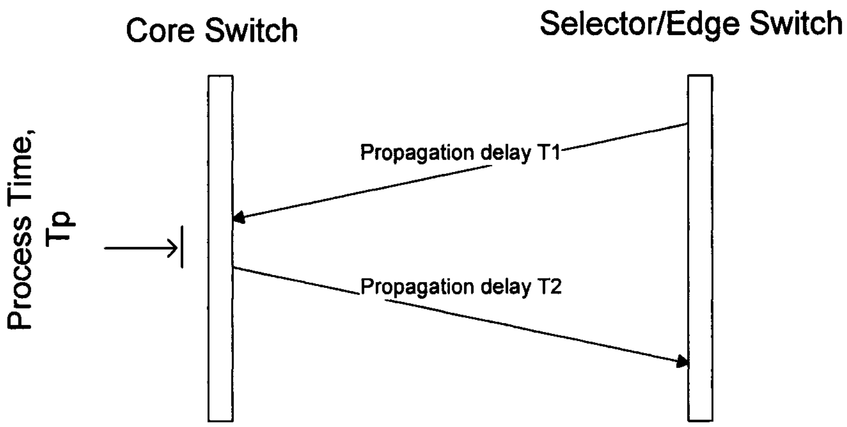


Figure 6-2 – Full Round Trip Time

T1 and T2 are almost fixed, moreover Tp is also fixed if the timeslot size is the same, therefore the only variable in this system is the time it actually arrived the Core Switch.

We will get the following as shown in Figure 6-3.

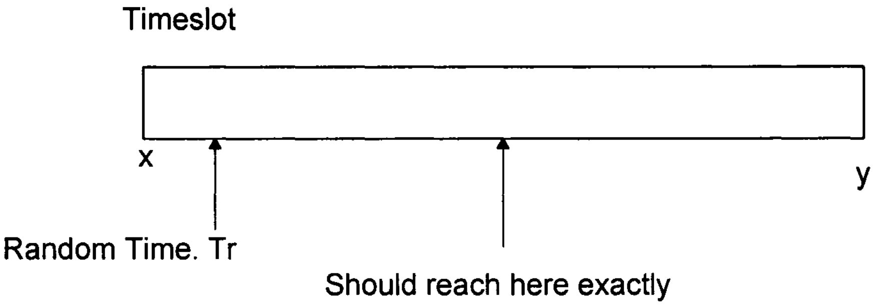


Figure 6-3 – First Random Iteration

Tr is a random variable to simulate the worst case instance of the algorithm. After this result is received another iteration will be needed. In the second iteration (Figure 6-4) instead of having the range between x and y, the new range is between Tr and y.

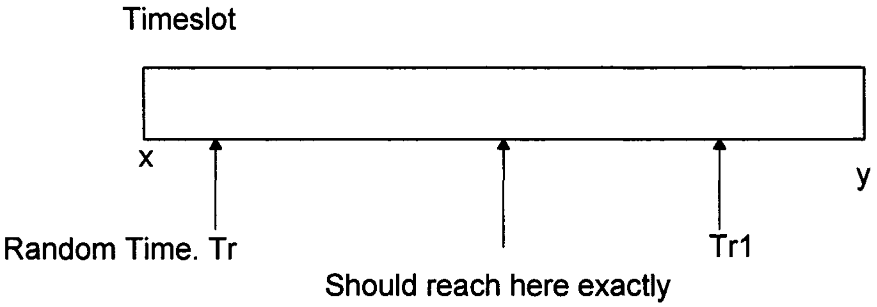


Figure 6-4 – Second Random Iteration

The third iteration is within the range of Tr and Trl and so on, until it reaches the required accuracy and value. Theoretically, this process is infinite, but since a specific accuracy is required the process has an end point and the program emulates the process exactly. It will take a range (x and y), the required value, and the accuracy needed. It then generates a random number (Tr), then it checks where does this value actually lays in the range. Then it makes the new range depending on the position of Tr. Since Tr is purely random, it actually may hit the required value the first time. To overcome this problem, the program was run 1000 times and then averaged to ensure that problems with the random variables were addressed. The program produces a graph of Tolerance Vs Number of Iterations, and the average number of iterations needed for a particular accuracy. In this project, the required tolerance is 100 ns, but for synchronization issues as this is mainly done during the initialization phase many cases will be taken depending on the system needed.

Prior to discussing the results the above is proven mathematically.

Let x_n= x at iteration n (random)

Range is X and Y => if Needed Value(T) $>x_1 > X$ => Range x_1 and Y

Meanwhile if $T<x_1<Y$ => Range is X and x_1

This goes on till $x_n \rightarrow T$

In summary, $x_1 + (x_2-x_1) + (x_3-x_2) \ldots x_n \rightarrow T$

Therefore the series is

$$X_1 + \sum_{n=1}^{\infty} (X_{n+1} - X_n) \longrightarrow T$$

The following graph Figure 6-5 shows the tolerance vs. number of iterations to reach 100ns accuracy. For solution 1 the average number of iterations to get 100ns accuracy is 8.034.

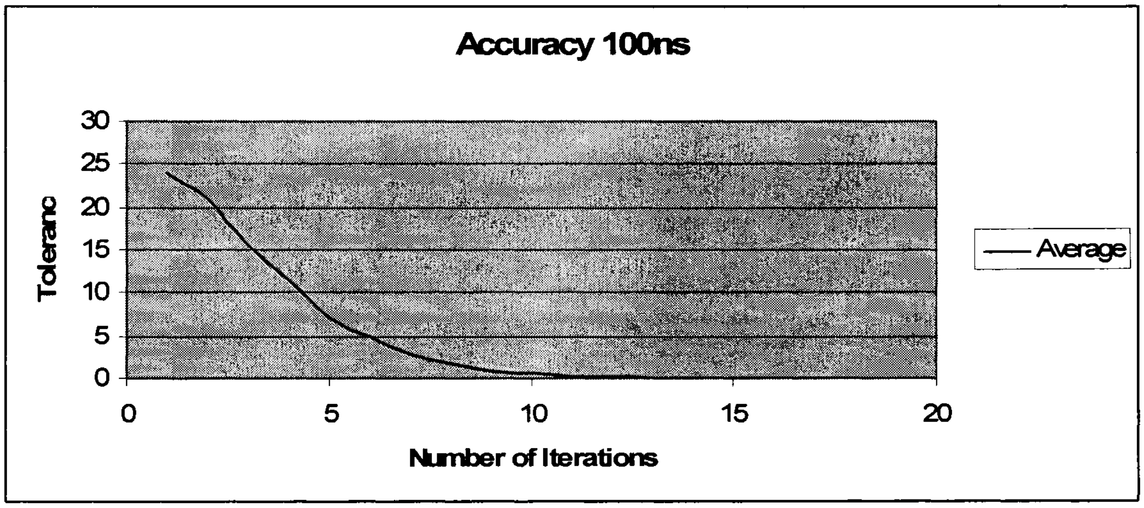


Figure 6-5 – Solution 1 – 100ns - Average Accuracy Graph

If a random run of the program is taken, the graph will be similar to Figure 6-6.

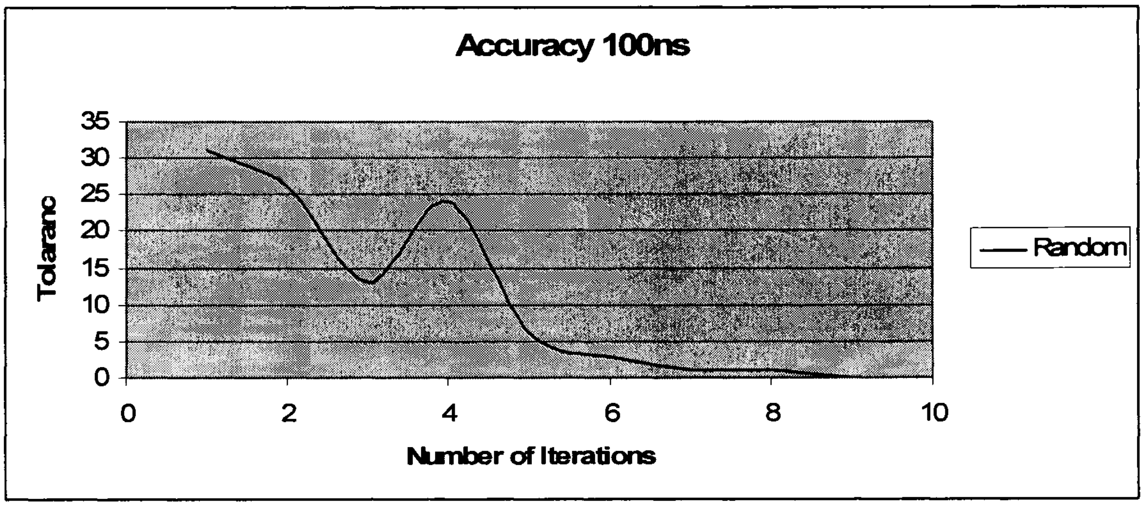


Figure 6-6 – Solution 1 – 100ns – Random Accuracy Graph

This raises the question as to why the iterations dip and then rise again at #4. The answer is that the algorithm takes a new range dependent on the random variable. For example, suppose the range is between 0 and 100, the needed value is 50 and the first random number is 40. The new range becomes 40 and 100. Then, if the second random variable is 80, its tolerance will be higher than the first random variable.

Figure 6-7 shows the difference between the maximum number of iterations and the average.

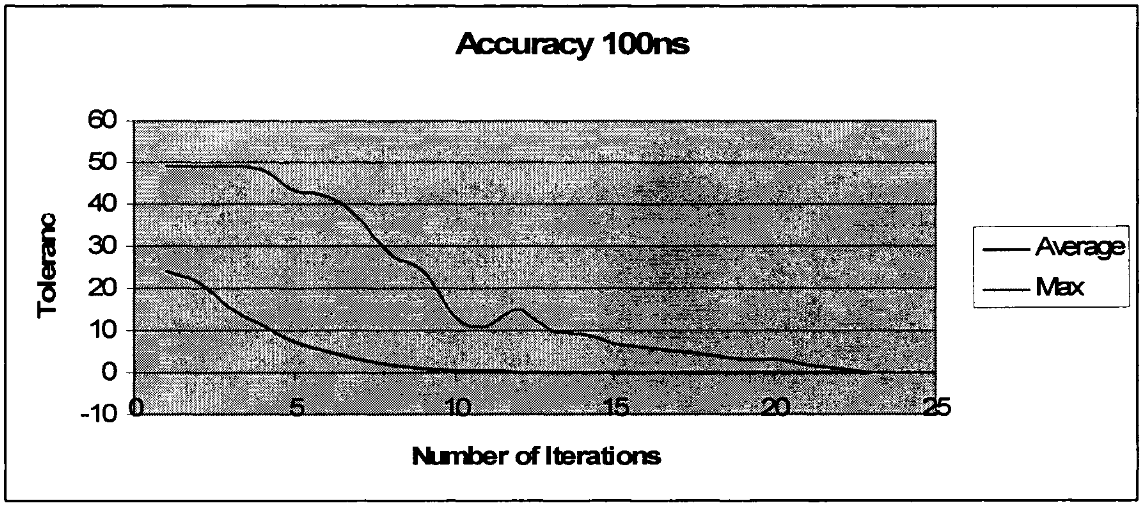


Figure 6-7 — Solution 1 – 100ns – Average Vs Max Accuracy Graph

Similarly modeling is performed for accuracies of 10ns and 1ns.

For the 10ns the average number of iterations is 12.789, moreover Figure 6-8 will illustrate more.

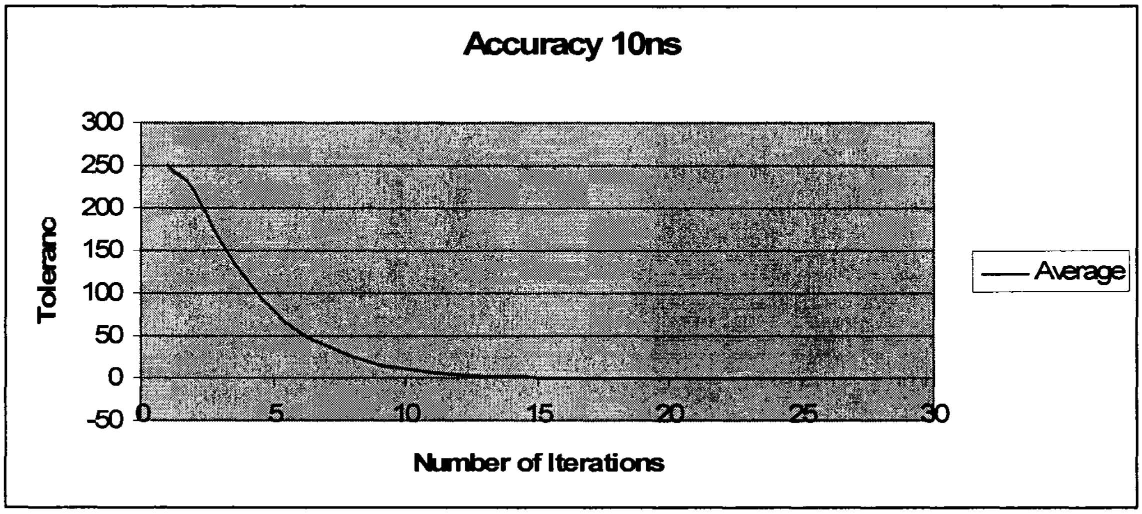


Figure 6-8 – Solution 1 – 10ns - Average Accuracy Graph

Taking a random run of the program in 10 ns will get Figure 6-9.

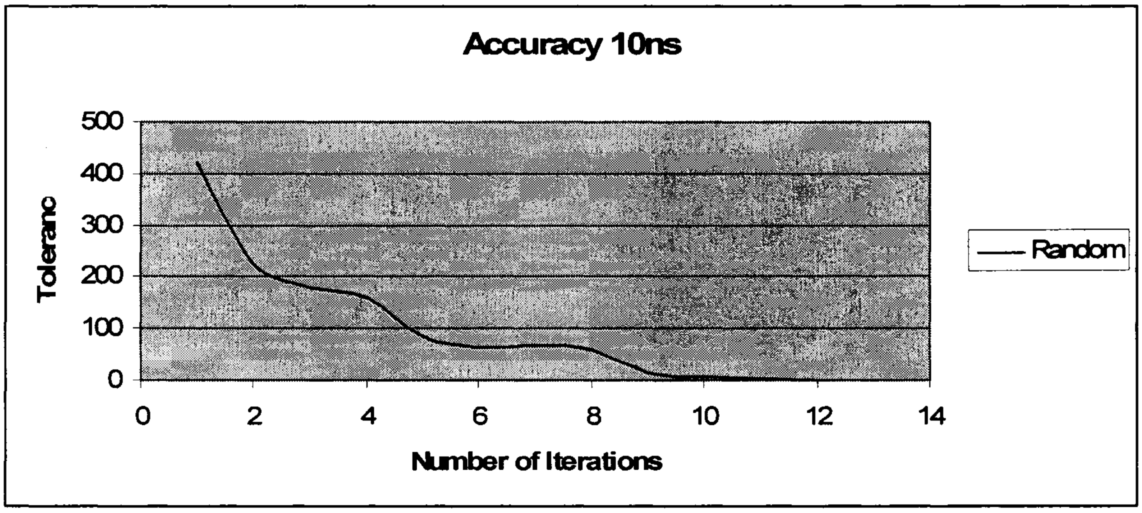


Figure 6-9 – Solution 1 – 10ns - Random Accuracy Graph

Figure 6-10 compares the Max and the Average values.

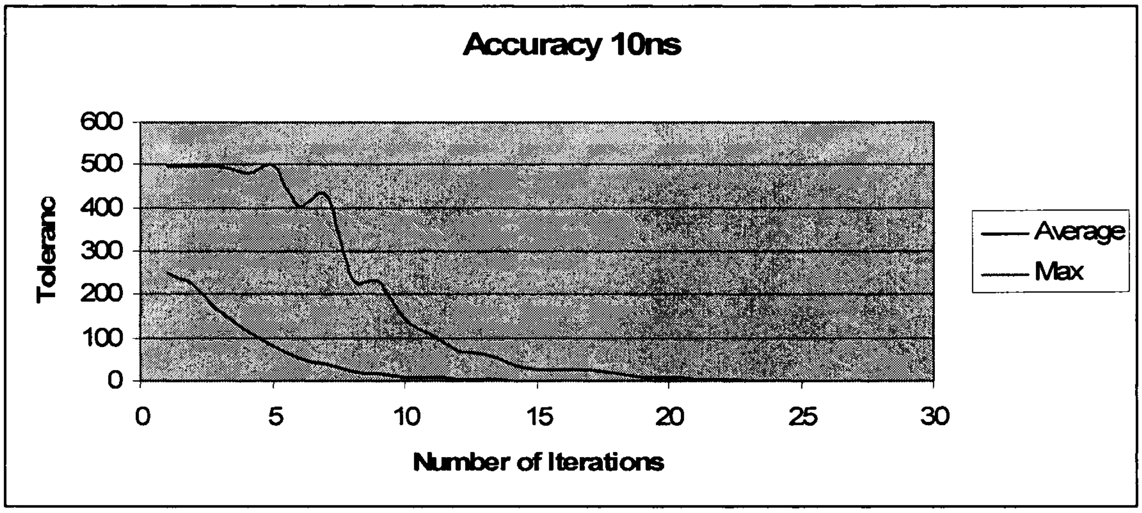


Figure 6-10 – Solution 1 – 10ns - Average Vs Max Accuracy Graph

The average number of iterations needed to get 1ns accuracy is 17.213. The graphs are shown in Figure 6-11.

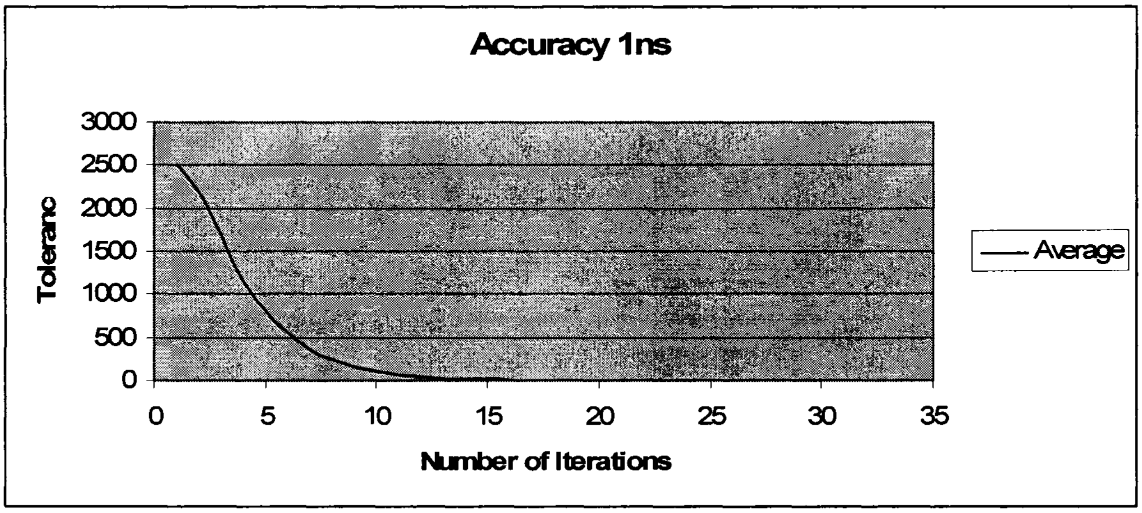


Figure 6-11 – Solution 1 – 1ns - Average Accuracy Graph

A random run for the 1ns tolerance will give Figure 6-12.

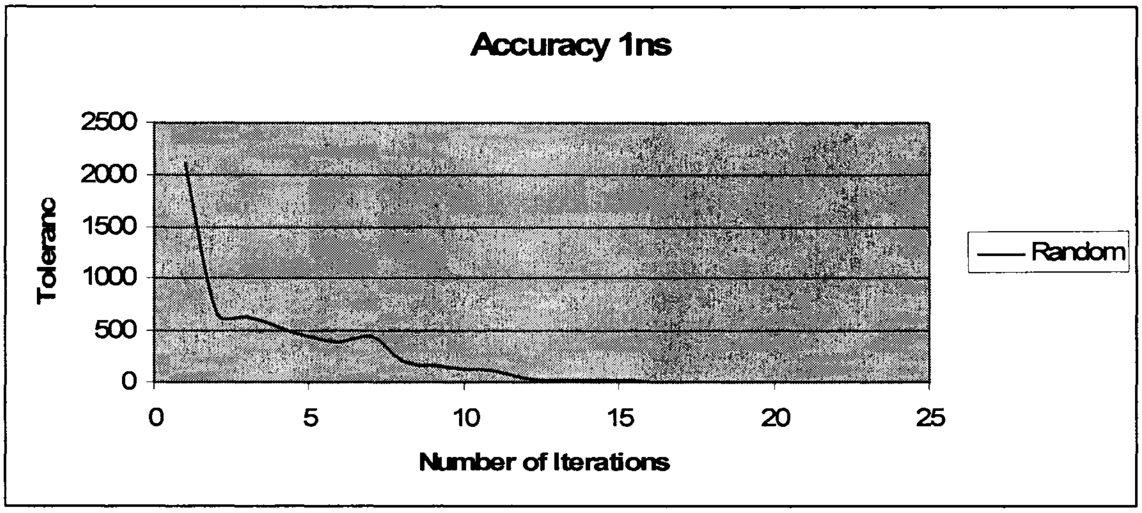


Figure 6-12 – Solution 1 – 10ns - Random Accuracy Graph

Finally Figure 6-13 is a comparison of the Average and the Max of 1ns accuracy.

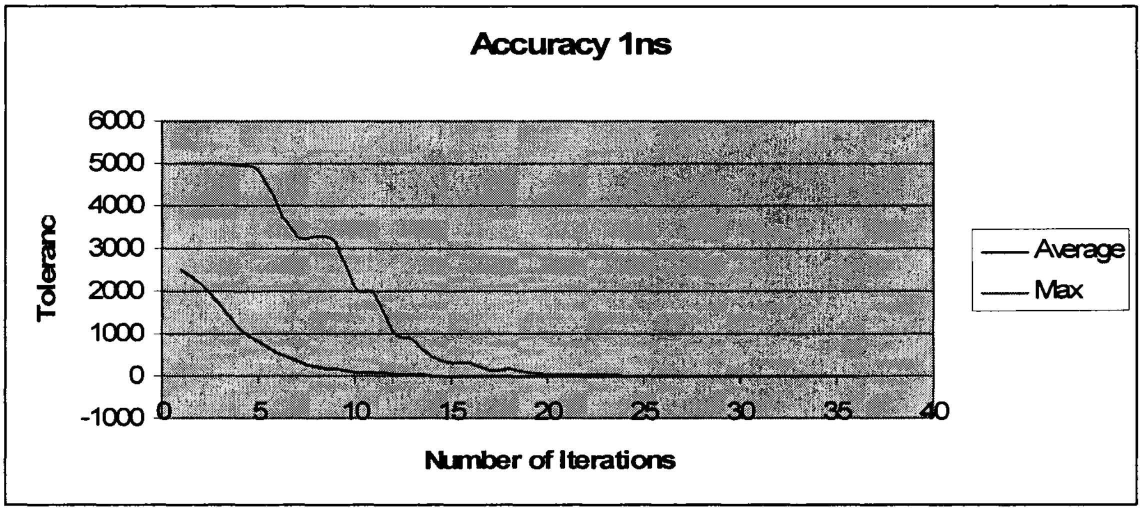


Figure 6-13 - – Solution 1 – 1ns - Average Vs Max Accuracy Graph

Figure 6-14 shows the comparison of the three accuracy standards

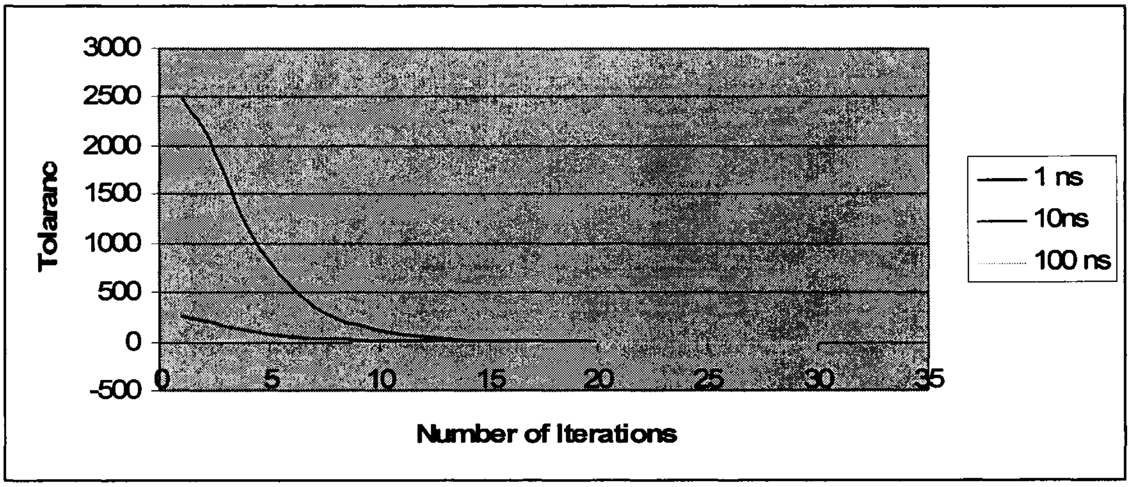


Figure 6-14 – Solution 1 – Comparing Different Accuracy Graph

6.2 Method 2

The following graphs will show the performance for solution 2. It is expected that solution 2 will perform better than solution 1, as it does not include propagation delays and other factors. The average number of iterations for 100ns accuracy is 5.937 and shown in Figure 6-15.

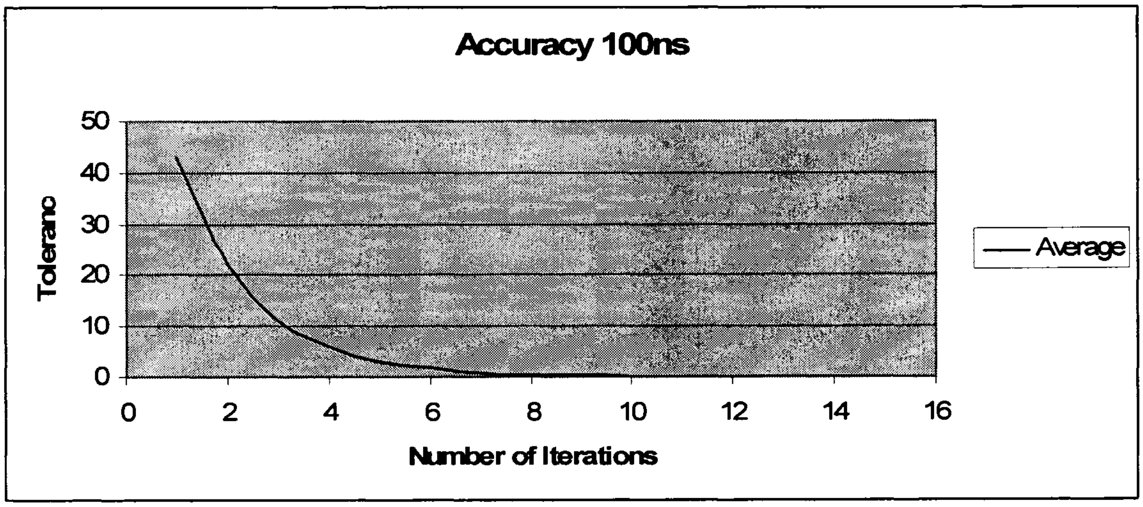


Figure 6-15 – Solution 2 – 100ns - Average Accuracy Graph

Taking random run into consideration the (Figure 6-16) will be shown

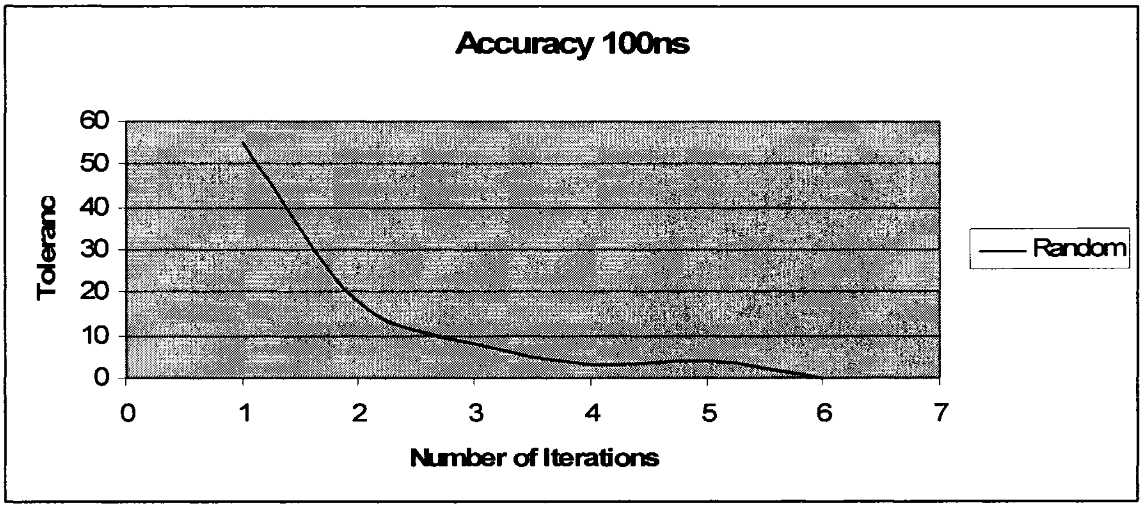


Figure 6-16 – Solution 2 – 100ns - Random Accuracy Graph

Finally (Figure 6-17) compares the average and max,

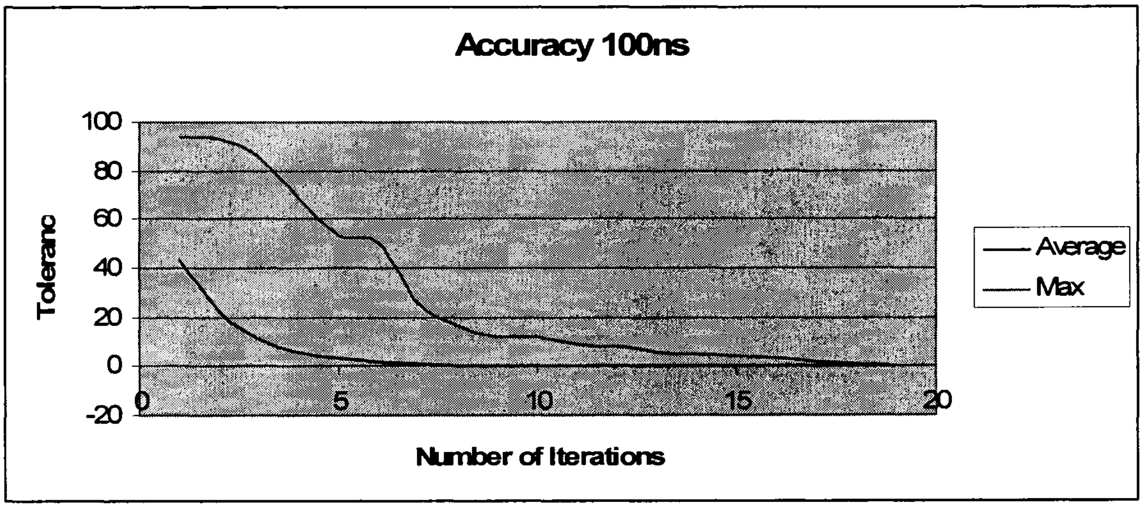


Figure 6-17 – Solution 2 – 100ns - Average Vs Max Accuracy Graph

A similar process is executed for 10nsec and 1nsec accuracy. For 10 ns second the average number of iterations for the solution 2 is 9.547. Figure 6-18 shows the comparison between the average of runs and the max runs.

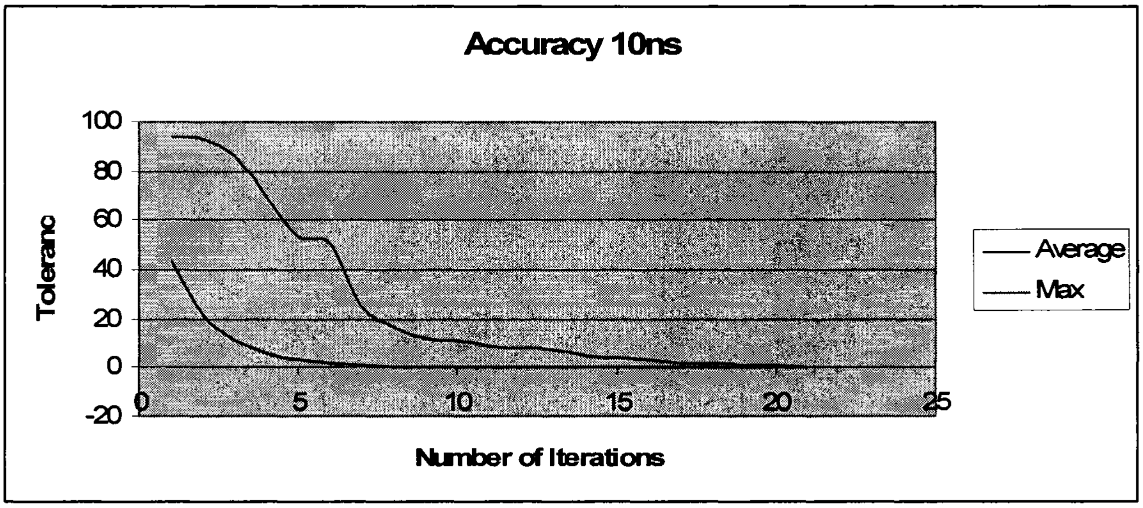


Figure 6-18 – Solution 2 – 10ns - Average Vs Max Accuracy Graph

The average time for 1 nsec is 13.627 and the graph for average and max is shown in Figure 6-19.

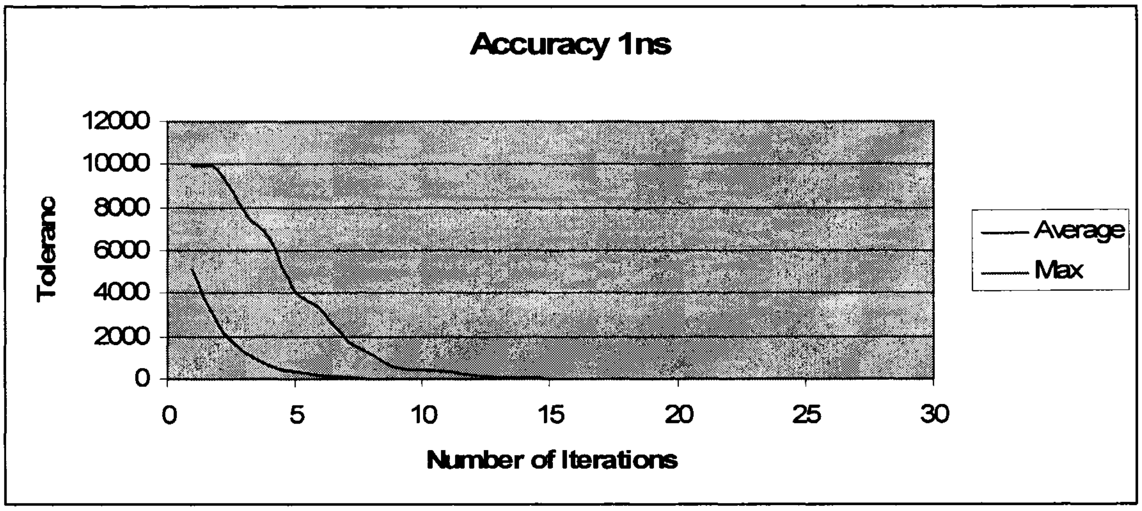


Figure 6-19 – Solution 2 – 1ns - Average Vs Max Accuracy Graph

6.3 GPS Method

Finally the performance analysis for the third solution (GPS clocks) will be explained. As mentioned earlier the tolerance for the GPS clocks is 100ns so more iterations should be done to tune to the required performance. The average number of iterations for 100ns for GPS clocks is 1.134. This means that most clocks will be aligned as they are already synchronized by GPS satellites on the first iteration. The average number of iterations for 10ns accuracy is 4.724 and finally for 1ns accuracy is 9.426. The analysis demonstrates that the GPS solution will perform better than others as they are already synchronized and solution 2 follows if GPS clocks are not available

6.4 Drifting of Clocks

The core switch should use the highest quality clock in terms of stability and accuracy because it is the most important clock in the network. Atomic clocks are the best kind of clocks. For example, a rubidium atomic could be used at the Core Switch and various types of clocks can be used for the Selector/Edge Switches. This is done to check the drifting rate of the clocks and how often updates should be sent so that synchronization is not lost over time.

There are two types of drifting: 1) Stability, or the ability of a clock to generate a time interval with constant value over a period of time and; 2) Accuracy, or how close the generated time interval is to the UTC time interval. Usually the accuracy of the clock gives its lifetime to produce it and is in years. This thesis compared the short term stability of the clocks. In this project, the network should be fully synchronized. That means that the drifting of clocks should be handled before it causes the system to drift outside its required tolerance. If it is left unsolved then this will lead in the loss of synchronization as explained earlier.

The following will show the update rates for different types of clocks to overcome drift. If a TCXO is used then an update should be sent every 1000 seconds. That is the maximum amount of time because after that it will drift and not all of the timeslots will be used efficiently. This is because its short-term stability is $1*10^{-9}$/second. Similarly this can be used for other types of clocks from the tables given above. Likewise if a Double Oven OCXO is used then updates should occur within 10000 seconds. Moreover a Double Oven BVA OCXO should have updates every $1 * 10^7$ seconds, which is roughly 12 days. When using atomic clocks, the rate of update messages is reduced because their stability is very high. When using the rubidium atomic clocks updates should be sent within 12 days. If cesium clocks are used, which are the most stable kind of clocks, then updates should be sent within 2 years. However, cesium clocks are very expensive and large. GPS clocks are synchronized with GPS satellites so their drifting rates are updated regardless.

7 Conclusion

The purpose of this thesis was to solve the problem of synchronization in photonic networks. An explanation of the full problem was provided in addition to the literature background of the problem and subtopics. It has been shown that this was possible with two different solutions depending on the given cases. Three solutions were outlined in chapter 5. First solution specifically solves the problem for equal sided propagation delays. Meanwhile solution 2 and 3 can have different propagation delay for the upstream and downstream. Moreover this thesis shows a performance analysis of the algorithm and shows that it converges with time and number of iterations. This might vary with the type of clock being used. Finally relative work like NTP was introduced; surely it was not entirely explained as it is a huge protocol

References, Sources and Bibliography

[1]. M. Coates, G. v. Bochmann, T. Hall, L. Mason, R.Vickers, O. Yang: "The Agile All Photonic Network: An Architectural Outline": Proc. Queen's Biennial Symposium on Communications 2004 pp.217-218

[2]. Stefano Bregni, "Synchronization of Digital Telecommunications Networks", John Wiley & Sons Ltd, 2002

[3]. M. Beshai, R. Vickers, "Agile optical-core distributed packet switch", U.S. Patent no. 6,486,983, 2002

[4]. M. Beshai, R. Vickers, "Methods and apparatus for distributed control of a multi-class network", U.S, Patent no. 6,404,735, 2002

[5]. M. Beshai, R. Vickers, "Self-configuring distributed switch", U.S. Patent no. 6,570,872, 2003

[6]. M. Beshai, R. Vickers, "State information and routing table updates in large scale data networks", U.S. Patent no. 6,744,775, 2004

[7]. David L. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis", *RFC-1305,* University of Delaware, 1992

[8]. David L. Mills, "Computer Network Time Synchronization : The Network Time Protocol", Taylor & Francis, 2006

[9]. Peter Rybaczyk, "Expert Network Time Protocol : an Experience in Time with NTP", Berkeley, 2005

[10]. Masami Kihara, Sadayasu Ono, Pekka Eskelinen "Digital Clocks for Synchronization and Communications", Boston, 2003

[11]. Tony Jones, "Splitting The Second : The Story of Atomic Time", Bristol, 2000

[12]. James Jespersen and Jane Fitz-Randolph, "from Sundials to Atomic Clocks : Understanding Time and Frequency", Washington D.C., 1999

[13]. F.G. Major, " The Quantum Beat : The Physical Principles of Atomic Clocks", Springer-Verlag, New York, 1998

[14]. Sachin Sapatnekar, "Timing", Kluwer Academic Publishers, 2004

[15]. Claude Audoin, Bernard Guinot, "The Measurement of Time : Time, Frequency and the Atomic Clock", Cambridge University, 2001

[16]. J. Viterbi, "Principles of Coherent Communications", New York: McGraw–Hill, 1966

[17]. Blanchard, "Phase-Locked Loops", New York, 1976

[18]. F. M. Gardner, "Phaselock Techniques", New York: John Wiley & Sons, 1979

[19]. R. E. Best, "Phase Locked Loops", New York: McGraw–Hill Book Company, 1984

[20]. *Frequency-Locked Loops, and Amplitude Control,* New York: John Wiley & Sons, 1990

[21]. E. A. Gerber and A. Ballato, "Precision Frequency Control", New York: Academic Press, 1985

[22]. S. Benedetto, E. Biglieri, V. Castellani, "Digital Transmission Theory". Englewood Cliffs, NJ, U.S.A, 1987

[23]. J. G. Proakis, "Digital Communications". New York, 1989

[24]. S. Haykin, "An Introduction to Analog and Digital Communications". New York: John Wiley & Sons, 1989

[25]. A. J. Viterbi, "Principles of Coherent Communications", New York: McGraw-Hill, 1966

[26]. W. C. Lindsey, "Synchronization Systems in Communications and Control", Englewood Cliffs, Prentice Hall Inc., 1972

[27]. J. Vig, "Quartz Crystal Resonators and Oscillators—A Tutorial", *Tech. Report SLCETTR-88-1,* US Army Electronics Technology and Devices Lab, July 1992.

[28]. Jacques Vanier, Claude Audoin, "The Quantum Physics of Atomic Frequency Standards", vol. 1, Bristol: Adam Hilger, 1989

[29]. E. Kaplan, "Understanding GPS: Principles and Applications", London, 1996

[30]. T. Beadles, "Introduction to GPS Applications", *http://ares.redsword.com/gps/*

[31]. P. H. Dana, B. Penrod, "The Role of GPS in Precise Time and Frequency Dissemination", *(reprinted from GPS World, July/August 1990),* http://www.bancomm.com/cgprole.htm

[32]. Russian Federation Ministry of Defense, Coordination Scientific Information Center, *Global Navigation Satellite System, GLONASS,* http://mx.iki.rssi.ru/SFCSIC/english.html

[33]. James Bao-yen Tsui, Fundamentals of global positioning system receivers: a software approach", New Jersey : John Wiley & Sons, 2005

[34]. D. Veitch, S. Babu, and A. Pàsztor, "Robust synchronization of software clocks across the Internet," in Proc. ACM SIGCOMM Conf. Internet measurement, Taormina, Sicily, Italy, 2004

[35]. R. T. Hofmeister, C. -L. Lu, M. -C. Ho, P. Poggiolini, and L. G. Kazovsky, "Distributed Slot Synchronization (DSS): A Network-Wide Slot Synchronization Technique for Packet-Switched Optical Networks," J. Lightwave Technol., vol. 16, no. 12, Dec. 1998, pp. 2109-2116