DOI: 10.21917/ijct.2012.0068

NETWORK PERFORMANCE FOR MULTI LINGUAL DATA TRANSMISSION

C.L. Brindha Devi[1] and P. Navaneethan[2]

[1]*Department of Computer Science, Arignar Anna Government Arts College for Women, India*
E-mail: clbrindhadevi@gmail.com
[2]*Department of Electrical and Electronics Engineering, PSG College of Technology, India*
E-mail: pnn@eee.psgtech.ac.in

Abstract

This paper compares different character encoding schemes used to encode the characters in different languages. A new character encoding protocol called PANDITHAM has been developed to encode the characters in different languages. The languages English and Tamil are taken for a case study and its performance under networking environment is compared with regard to PANDITHAM, Unicode and UTF-8 encodings. This study has proved that PANDITHAM is optimal for all languages as it reduces the network congestion.

Keywords:

Multilingual, PANDITHAM, Unicode, UTF-8

1. INTRODUCTION

Internet traffic is the flow of data around the Internet. It includes web traffic, which is the amount of data that is related to the World Wide Web, along with the traffic from other major uses of the Internet, such as electronic mail and peer-to-peer networks. Some companies offer advertising schemes that in return contribute to increase in web traffic. The World Wide Web has become a major channel for information service. There are web pages in almost every popular language including various European, Asian, and Middle East languages [2]. While approximately 70% of web content is in English, the number of native English speakers constitutes only 36.5% of the world's online population [5]. The rapidly accelerating trend of globalization of businesses and the success of e-Governance solutions require data to be stored and manipulated in many different natural languages.

2. CHARACTER ENCODING

In computers and in data transmission between them, data is internally presented as octets. Octets are called as bytes. A character is thought of as the smallest component of written language that has a semantic value. The set of all the characters in a language is called a Repertoire [3]. Each character in the repertoire is assigned a unique numerical code called Code Position. A character encoding defines how sequences of numeric codes are presented as sequences of octets. For many years, Americans have transmitted data using the ASCII character set. But ASCII is inadequate in handling the characters of all other languages. Different countries have adopted different techniques for exchanging text in different languages, making it difficult to exchange data in an interconnected world. There are many character encoding systems like ASCII, Unicode, EBCDIC, ISO-8859 [4] etc. This paper compares the encoding of Multilingual characters using Unicode, UTF-8 and PANDITHAM (A Protocol for ApplicatioNs Development In THAmizh and Multilingual Computing) and as a case study the languages Tamil and English are considered.

2.1 UNICODE CHARACTER ENCODING

Unicode [8] is a universal font encoding scheme, designed to cover all world languages. It is a 16-bit scheme with over 65500 slots to assign to various languages. Each language (except few like Chinese) is given a 128-slot block. All Indic languages are allocated 128-slots each. Assignment of characters to specific slots within this block is based on ISCII (Indian Script Code for Information Interchange) [9] scheme, that uses Devanagari as the basic reference language. Refer to Table.1 for Tamil characters in Unicode. Thus the vowels, consonants and their modifiers of each Indic language appear at the same slot. "Ka" of Tamil and Telugu are separated by the same 128 slots, greatly facilitating programming.

The character set in Tamil language shall be categorized into frequently used Tamil characters and infrequently used Tamil characters. The language contains a total of 313 (247 + 66) characters. The frequently used set of Tamil characters is divided into consonants, vowels and combined characters. Tamil language has 12 vowels (Uyir Eluthukkal) and 18 consonants (Mei Eluthukkal). The vowels come after consonants and combine with consonants to form the composite consonants. This way, the combination of 12 vowels and 18 consonants form 216 composite consonants. The coding for Tamil is not as per the Tamil alphabetical (Akkara Varisai) order. Since ISCII is the base for Unicode, it needs 2 bytes for encoding Uyir Eluthukkal [அ... ஔ] and Akkara Mei Eluthukkal [க...ன], and 4 bytes to encode Mei Eluthukkal [க்...ன்] and Uyir Mei Eluthukkal [கா கி கு…கௌ]. The English language has 26 characters. All the characters can be stored in the given slot. So Unicode uses 2 bytes to encode these English characters. Consider for example, the word "மாதேஸ்வரன் R." in Tamil. When this word is encoded using Unicode it needs 26 bytes.

Name:	மா	தே	ஸ்	வ	ர	ன்	R	.
	4	4	4	2	2	4	2	2	2

No of Bytes needed: 26 bytes

The corresponding Unicode sequence would be,

ம ா த ே ஸ ் வ ர ன ் . R .

0BAE, 0BBE : 0BA4, 0BC7 : 0BB8, 0BCD : 0BB5 : 0BB0 : 0BA9, 0BCD : 0020 : 0052 : 002E.

2.2 UTF- 8 CHARACTER ENCODING

UTF stands for Unicode Transformation Format. The '8' means that it uses a series of 8-bits to represent a character. The number of bytes needed to represent a character varies from 1 to 6. Most software is not designed to handle 16-bit or 32-bit characters. Therefore, a special format called UTF-8 was developed so as to encode the international characters in a format more easily handled by existing programs and libraries. UTF-8 is a variable-width encoding scheme. The characters having Hex value between 0 to 0x7f encode themselves as a single byte, while characters with larger values are encoded into 2 to 6 bytes of information. For use in web pages, the Unicode based text must be stored in UTF-8 format. UTF-8 encodes characters based on the Hex value.

UTF-8 [7] needs 3 bytes for encoding Uyir Eluthukkal [அ...ஔ] and Akkara Mei Eluthukkal [க...ன], and 6 bytes to encode Mei Eluthukkal [க்...ன்] and Uyir Mei Eluthukkal [கா கி கு...கௌ]. The characters in the English language can be encoded using a single byte as its Hex value is between 0x00 and 0x7F.

Consider for example the word "மாதேஸ்வரன் R.". When this word is encoded using UTF-8, it needs 33 bytes.

Name:	மா	தே	ஸ்	வ	ர	ன்	R	.
	6	6	6	3	3	6	1	1	1

No. of Bytes needed : 33 bytes

The requirement of 6 bytes for மா can be explained as follows:

மா in Tamil would have to be encoded in Unicode as ம followed by ா, whose 16-bit Unicode representations are respectively, 0BAE and 0BBE. Each of these 16-bit words in UTF-8 format would demand 3 bytes each, as these 2 bytes are packed in certain pre-determined bit positions [10]. For example, 0BAE in UTF-8 would be in the form of a Head Byte followed by one or more Tail Bytes. Tail Bytes would always start with the MSBs as "10", whereas the Head Byte will have a run of 1's followed by a 0. The run length of 1's determined the total number of Bytes needed [7]. The free space from these bytes is used to pack the Unicode value. For example, the Unicode 0x0BAE is packed in UTF-8 format as illustrated in Fig.1. Hence, the UTF-8 representation for ம is = E0, AE, AE (0BAE). Likewise another 3 bytes are required for ா (i.e. 0BBE).

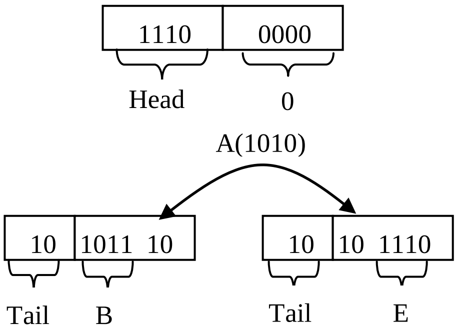

Fig.1. Packing of 16-bit Unicode in UTF-8 format

Table.1. UNICODE – THAMIZH TABLE (0B80– 0BFF)

	0B80	0B90	0BA0	0BB0	0BC0	0BD0	0BE0	0BF0
0		ஐ		ர	◌ீ			௰
1				ற	◌ு			௱
2		ஒ		ல	◌ூ			௲
3	ஃ	ஓ	ண	ள				
4		ஔ	த	ழ				
5	அ	க		வ				
6	ஆ				◌ெ			
7	இ			ஷ	◌ே	◌ௗ	௧	
8	ஈ		ந	ஸ	◌ை		௨	
9	உ	ங	ன	ஹ			௩	
A	ஊ	ச	ப		◌ொ		௪	
B					◌ோ		௫	
C		ஜ			◌ௌ		௬	
D					◌்		௭	
E	எ	ஞ	ம	◌ா			௮	
F	ஏ	ட	ய	◌ி			௯	

2.3 PANDITHAM

In this coding scheme, more appropriately a protocol, apart from consonants and vowels, a composite phonetic letter is also given a code. Thamizh language is thought of as made of two logical languages, namely, pure Thamizh and Grantha (vadamozhi) Thamizh. Refer to Table.2 and Table.3 for Thamizh phonetic character encoding. Since, the number of phonetically differing characters in a language are likely to exceed 128, some control characters of ASCII are used as well. This calls for a Protocol rather than a simple Encoding scheme. This section reviews a relevant protocol namely, PANDITHAM [1]. It is an 8-bit character oriented protocol.

2.3.1 Basic Principles:

Most of the languages have lexical order associated with their letters. Hence, for each letter one can associate the lexical order number itself as part of coding the letter. Refer to Table.2 for Thamizh letter coding in PANDITHAM. In general, a Multilingual string may have a combination of letters from more than one language and hence, the language aspect is first standardized, namely, language codes. These assignments are meant for the sake of illustrations.

LANGUAGES	**CODES**
ASCII	05H(ASC)
THAMIZH1	08H(TM1)
THAMIZH2	09H(TM2)
TELUNGU	0AH(TLU)

We now look at standardization and meaning of PANDITHAM Control and/or Punctuation characters. Start of the string: The control character DLC (02H), shall be followed by the code of the language of the ensuing string, where DLC stands for Default Language Code.

Table.2. PANDITHAM Thamizh Table (பண்டிதம் தமிழ்)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SP	DLC	MLC	!	?	,	-	.	அ	ஆ	இ	ஈ	உ	ஊ	எ
1	ஏ	ஐ	ஒ	ஓ	ஔ	ஃ	க	கா	கி	கீ	கு	கூ	கெ	கே	கை	கொ
2	கோ	கௌ	க்	ங	ஙா	ஙி	ஙீ	ஙு	ஙூ	ஙெ	ஙே	ஙை	ஙொ	ஙோ	ஙௌ	ங்
3	ச	சா	சி	சீ	சு	சூ	செ	சே	சை	சொ	சோ	சௌ	ச்	ஞ	ஞா	ஞி
4	ஞீ	ஞு	ஞூ	ஞெ	ஞே	ஞை	ஞொ	ஞோ	ஞௌ	ஞ்	ட	டா	டி	டீ	டு	டூ
5	டெ	டே	டை	டொ	டோ	டௌ	ட்	ண	ணா	ணி	ணீ	ணு	ணூ	ணெ	ணே	ணை
6	ணொ	ணோ	ணௌ	ண்	த	தா	தி	தீ	து	தூ	தெ	தே	தை	தொ	தோ	தௌ
7	த்	ந	நா	நி	நீ	நு	நூ	நெ	நே	நை	நொ	நோ	நௌ	ந்	ப	பா
8	பி	பீ	பு	பூ	பெ	பே	பை	பொ	போ	பௌ	ப்	ம	மா	மி	மீ	மு
9	மூ	மெ	மே	மை	மொ	மோ	மௌ	ம்	ய	யா	யி	யீ	யு	யூ	யெ	யே
A	யை	யொ	யோ	யௌ	ய்	ர	ரா	ரி	ரீ	ரு	ரூ	ரெ	ரே	ரை	ரொ	ரோ
B	ரௌ	ர்	ல	லா	லி	லீ	லு	லூ	லெ	லே	லை	லொ	லோ	லௌ	ல்	வ
C	வா	வி	வீ	வு	வூ	வெ	வே	வை	வொ	வோ	வௌ	வ்	ழ	ழா	ழி	ழீ
D	ழு	ழூ	ழெ	ழே	ழை	ழொ	ழோ	ழௌ	ழ்	ள	ளா	ளி	ளீ	ளு	ளூ	ளெ
E	ளே	ளை	ளொ	ளோ	ளௌ	ள்	ற	றா	றி	றீ	று	றூ	றெ	றே	றை	றொ
F	றோ	றௌ	ற்	ன	னா	னி	னீ	னு	னூ	னெ	னே	னை	னொ	னோ	னௌ	ன்

Table.3. PANDITHAM Grantha Table (*பண்டிதம் தமிழ் வடமொழி*)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SP	DLC	MLC	!	?	,	-	.							
1																
………….																
B															ஹ	ஹா
C	ஹி	ஹீ	ஹு	ஹூ	ஹெ	ஹே	ஹை	ஹொ	ஹோ	ஹௌ	ஹ்	ஷ	ஷா	ஷி	ஷீ	ஷு
D	ஷூ	ஷெ	ஷே	ஷை	ஷொ	ஷோ	ஷௌ	ஷ்	ஸ	ஸா	ஸி	ஸீ	ஸு	ஸூ	ஸெ	ஸே
E	ஸை	ஸொ	ஸோ	ஸௌ	ஸ்	ஜ	ஜா	ஜி	ஜீ	ஜு	ஜூ	ஜெ	ஜே	ஜை	ஜொ	ஜோ
F	ஜௌ	ஜ்	க்ஷ	க்ஷா	க்ஷி	க்ஷீ	க்ஷு	க்ஷூ	க்ஷெ	க்ஷே	க்ஷை	க்ஷொ	க்ஷோ	க்ஷௌ	க்ஷ்	ஸ்ரீ

Change of Language:

The change can be in two different ways.

Way 1: The switching of language is such that at least 2 characters are there in the new language. In this case, once again DLC is used in a similar way.

Way 2: The switching is momentary in nature; i.e., only one character temporarily changes to a new language, and the rest follows the previous language itself. To denote such an occurrence, we shall use the control character MLC (03h) to be followed by the language code, where MLC stands for Momentary Language Code.

- Termination of the string: NULL character (i.e., 00H) shall serve the purpose.
- Frequently used de-limiting characters: Out of the 256 characters, we have already made use of three control characters, namely, DLC, MLC and NULL. The rest can in fact be used to codify letters of various languages. As an instance, the 247 Thamizh letters shall be pushed in, and the remaining 6 can be used for codifying frequently used punctuation mark.

2.3.2 Design of Language Related Databases:

Any multilingual data processing should be based on the language and not on fonts, which are vulnerable to change. To facilitate this, a language database is to be maintained. The language database consists of details like unique language code, language name and weight. The languages can be classified into 3 categories based on the amount of storage requirements. The first category comprises of languages like English, which occupy single byte/character. The best example of second category would be Japanese language, which requires 2 bytes/character. The language Tamil comes under the third category, whose storage requirement on the average lies in between 1 and 2 bytes/character (on the average 1.1 bytes/character). This extra 0.1 is basically due to the presence of infrequently used Tamil characters (i.e., Grantha characters).

ISSN: 2229-6948(ONLINE) ICTACT JOURNAL ON COMMUNICATION TECHNOLOGY, MARCH 2012, VOLUME: 03, ISSUE: 01

Structure of Databases

The structure of the language related databases are given below:

```
Type Fonts
fontCode As Int              // Primary Key
fontName As String           // Font name in English
fontNamePtx As mlString      // Font name in that language as
                             //   PANDITHAM text
langCode As Byte             // Foreign Key & code denotes the
                             // language to which this font belongs to
langCodeSpouse As Byte       // Foreign Key and this code if 00h
                             // would mean that this font doesn't
                             // serve any Spouse Language
End Type

Type Language
langCode As Byte             // Primary Key
langCodeSpouse As Byte       // Foreign key 00h No Spouse
langName As String           // Name in English
langNamePtx As mlString      // Language name in that language
                             // as PANDITHAM text
langWeight As Byte           // To facilitate language based
                             // Sorting
bytesPerChar As Byte         //  1 Byte for English, Tamil etc.
                             //  2 Bytes for Japanese, Chinese etc.
OffsetFont As Int            // To facilitate translating
                             // PANDITHAM character value
                             //into 16-bit Font value
defaultFontCode As Int       // Foreign Key & text in this
                             // language will get displayed in this
                             // font, unless specified otherwise
End Type
```

Fig.2 gives the allocation of spaces for languages like English, Tamil and Grantha in the Muhil Font. It is also true of Aruvi font, which is compatible with Muhil, in the same way as the Arial is to Times New Roman.

In Tamil, it needs only 1 byte to encode Uyir Eluthukkal [அ...ஔ], Akkara Mei Eluthukkal [க...ன], Mei Eluthukkal [க்...ன்] and Uyir Mei Eluthukkal [கா கி கு...கௌ]. Moreover, the characters in English (ASCII) language are also encoded using a single byte only. Consider for example the word "மாதேஸ்வரன் R.".

When this word is encoded using PANDITHAM, it needs 16 bytes.

Name: மாதேஸ்வரன் R.

DLC TM1 மா தே MLC TM2 ஸ் வ ர ன் SP DLC ASC R . NULL

where, SP - Blank Space. No. of Bytes needed: 16 bytes

02, 08, 8C, 6B, 03, 09, E4, BF, A5, FF, 20, 02, 05, 52, 2E, 00 (in Hex).

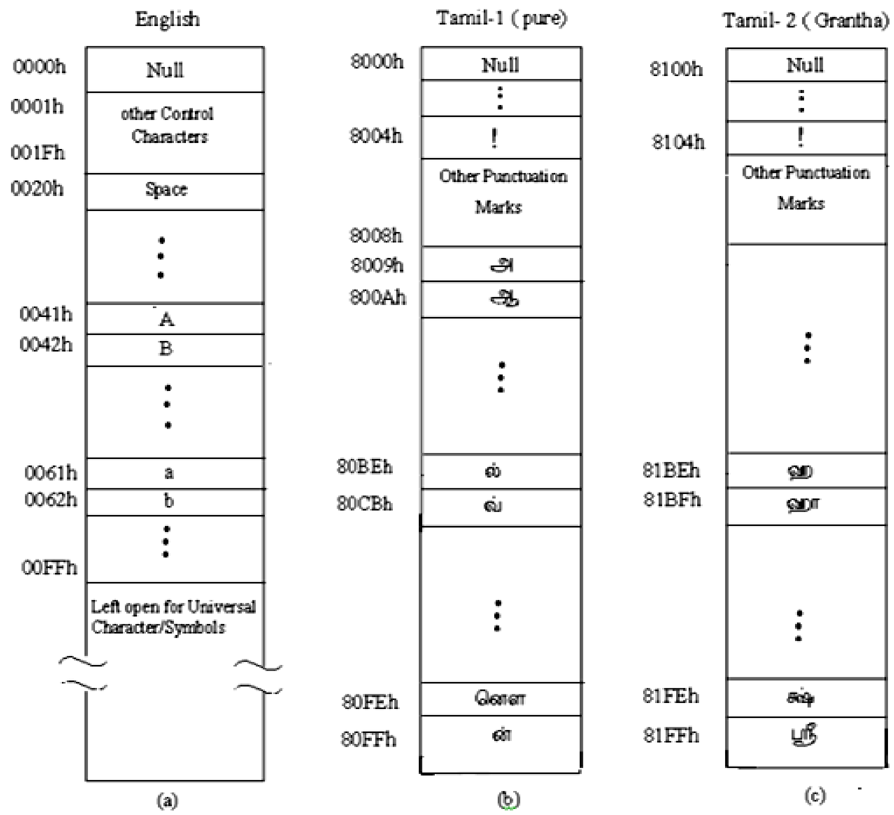

Fig.2. Muhil Font (a) Allocation for English (b) Allocation for Tamil-1 (Pure) (c) Allocation for Tamil-2 (Grantha)

In a similar way, the word "Madheswaren R." when encoded using PANDITHAM, it needs 17 bytes.

Name: Madheswaren R.

DLC ASC M a d h e s w a r e n SP R. NULL

where, SP - Blank Space. No. of Bytes needed: 17 bytes

02 05 4D 61 64 68 65 73 77 61 72 65 6E 20 52 2E 00 (in Hex).

3. COMPARISON

Consider for example, the multilingual string "மாதேஸ்வரன் R.". Let the word be encoded using Unicode. Since ISCII is the base for Unicode, it needs 2 bytes for encoding Uyir Eluthukkal [அ...ஔ] and Akkara Mei Eluthukkal [க ... ன] , and 4 bytes to encode Mei Eluthukkal [க்... ன்] and Uyir Mei Eluthukkal [கா கி கு... கௌ].

Name: மா தே ஸ் வ ர ன் R .
4 4 4 2 2 4 2 2 2

No. of Bytes needed : 26 bytes

When one wants to transfer a file containing Tamil text using Unicode, the numbers of packets generated are:

(Uyir Eluthukkal * 2 + Akkar Mei Eluthukkal * 2 + Mei Eluthukkal * 4 + Uyir Mei Eluthukkal * 4) / packet size

The English language has 26 characters. Unicode uses 2 bytes to encode these English characters.

When one wants to transfer a file containing English text using Unicode, the numbers of packets generated are:

(Total size of the file * 2) / Packet size

Let this word be encoded using UTF-8. UTF-8 needs 3 bytes for encoding Uyir Eluthukkal and Akkara Mei Eluthukkal and 6 bytes to encode Mei Eluthukkal and Uyir Mei Eluthukkal.

Name: மா தே ஸ் வ ர ன் R .
6 6 6 3 3 6 1 1 1

No. of Bytes needed: 33 bytes.

When one wants to transfer a file containing Tamil text using UTF-8, the numbers of packets generated are:

(Uyir Eluthukkal * 3 + Akkar Mei Eluthukkal * 3 + Mei Eluthukkal * 6 + Uyir Mei Eluthukkal * 6) / Packet size

The characters in the English language can be encoded using a single byte as its Hex value is between 0x00 and 0x7F.

When one wants to transfer a file containing English text using UTF-8, the numbers of packets generated are:

(Total size of the file * 1) / Packet size

Let this word be encoded using PANDITHAM.

மாதேஸ்வரன்R .

DLC TM1 மா தே MLC TM2 ஸ் வ ர ன் SP DLC ASC R . NULL

where, SP - Blank Space. No. of Bytes needed: 16 bytes

When one wants to transfer a file containing Tamil text using PANDITHAM, the numbers of packets generated are:

((Uyir Eluthukkal * 1 + Akkar Mei Eluthukkal * 1 + Mei Eluthukkal * 1 + Uyir Mei Eluthukkal * 1) + 2) / Packet size

When one wants to transfer the file containing English text using PANDITHAM, the numbers of packets generated are:

(Total size of the file * 1 +2) / Packet size

i.e., PANDITHAM takes 1 byte, on the average, to encode each English character.

When the same data is encoded with PANDITHAM, Unicode and UTF-8 it is found that PANDITHAM takes less number of bytes than the other two. This paper studies the amount of congestion that the network would be subjected to depending on the various Multilingual Information Encoding Schemes.

4. CONGESTION STUDY UNDER DIFFERENT MULTILINGUAL INFORMATION REPRESENTATION

When the same data is encoded with PANDITHAM, Unicode and UTF-8 it is found that PANDITHAM takes less number of bytes than the other two. This section studies the amount of congestion [9] that the network would be subjected to depending on the various Multilingual Information Encoding Schemes.

NS-2 simulations have been carried out to test the performance of Unicode, UTF-8 and PANDITHAM by sending a text file from source to destination using different Topologies, Queue sizes and Bandwidth etc., The sample topologies are shown in the Fig.3 and Fig.4, respectively. For each TCP [10] agent a new FTP application is defined. Each source produces a file of size 1 lakh characters to transmit using PANDITHAM, Unicode, UTF-8 encoding with different Topologies, Bandwidth and Queue size.

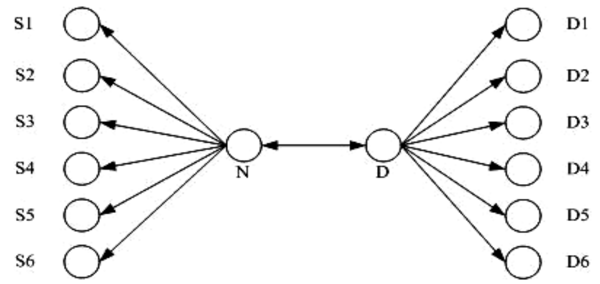

Fig.3. A Simple Regular Topology

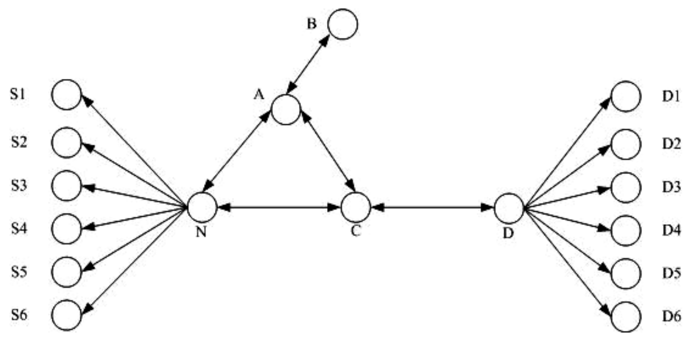

Fig.4. An Irregular Topology

4.1 STUDY OF NUMBER OF PACKETS TRANSMITTED

4.1.1 Tamil Text:

In general, consider that a file with one lakh Tamil Characters is transmitted using a propagation delay of 10 ms, and with a buffer capacity of the queue as 50 packets, then the number of packets transmitted and retransmitted with 1 Mbps of bandwidth using PANDITHAM, UNICODE and UTF-8 are shown in Fig.5 and Fig.6, respectively.

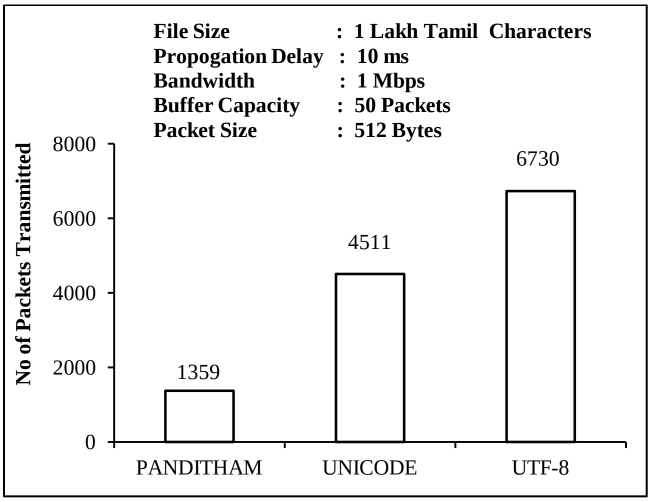

Fig.5. Packets Transmitted with Bandwidth = 1 Mbps

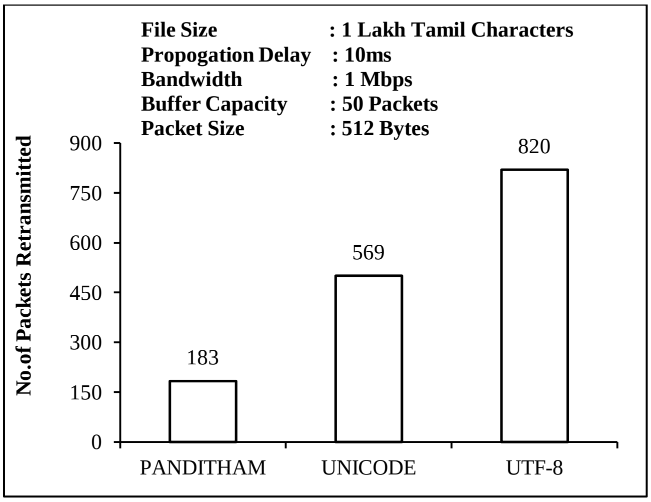

Fig.6. Packets Retransmitted with Bandwidth = 1 Mbps

This shows that PANDITHAM has transferred fewer packets for Tamil Characters compared to Unicode and UTF-8. This result proves that PANDITHAM format demands 'n' times less number of packets than Unicode and UTF-8 where n > 3. In the next setup, the buffer capacity is changed from 50 packets to 100 packets and the bandwidth from 1 Mbps to 10 Mbps and the corresponding comparison charts are shown in Fig.7 and Fig. 8, respectively.

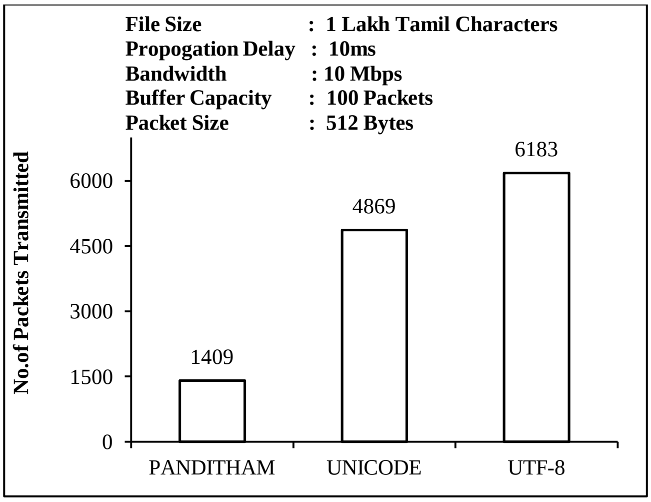

Fig.7. Packets Transmitted with Buffer Capacity = 100 packets

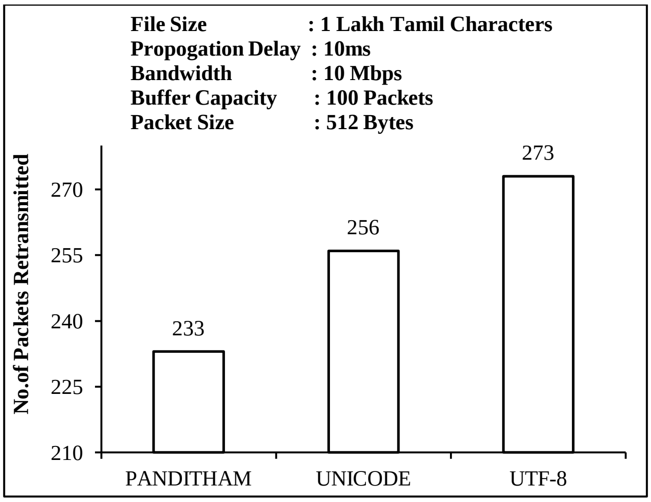

Fig.8. Packets Retransmitted with Buffer Capacity = 100 packets

Subsequently, the propagation delay alone is changed from 10 ms to 20 ms, and Fig.9 and Fig.10 depict the respective comparisons.

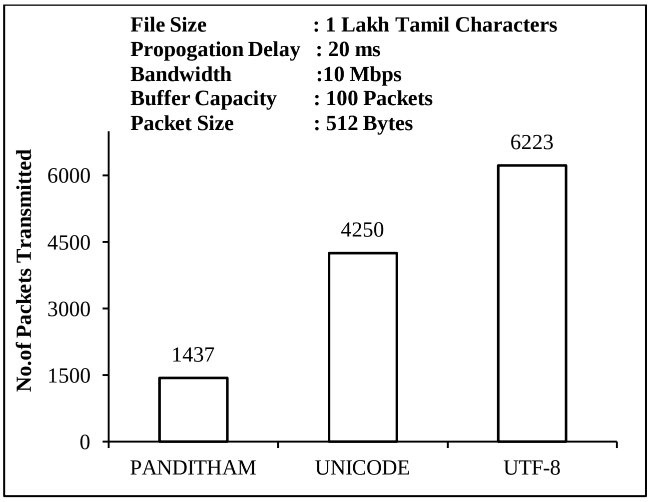

Fig.9. Packets Transmitted with Propagation Delay = 20 ms

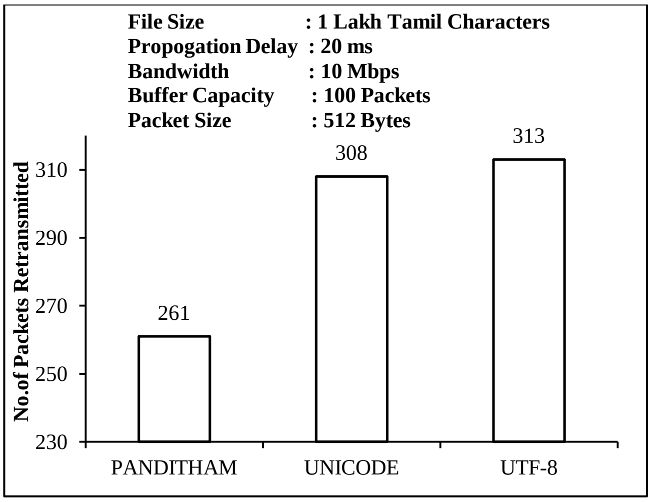

Fig.10. Packets Retransmitted with Propagation delay = 20 ms

From the analysis done, it is inferred that PANDITHAM has taken fewer number of packets to transfer 1 lakh Tamil characters compared to Unicode and UTF-8. When the traffic is less, more packets are transmitted in less amount of time, and this will subsequently reduce the number of retransmissions and reduce the network congestion, as well.

4.1.2 English Text:

A text file containing only English characters of an average size of “N” is transmitted using PANDITHAM, Unicode, UTF-8 encoding with different Topologies, Bandwidth and Queue size. When a file with 1 lakh English Characters is transmitted using the propagation delay of 10 ms and with a buffer capacity of the queue as 50 packets, then the number of packets transmitted and retransmitted with a bandwidth of 1 Mbps using PANDITHAM, UNICODE and UTF-8 are shown in Fig.11 and Fig.12, respectively. This shows that PANDITHAM and UTF-8 have transferred same number of packets for English Characters. These results prove that PANDITHAM and UTF-8 formats demand almost 2 times less number of packets than Unicode. In the next setup, the buffer capacity is changed from 50 packets to 100 packets and the bandwidth from 1mbps to 10mbps and the corresponding comparison charts are shown in Fig.13 and Fig.14, respectively.

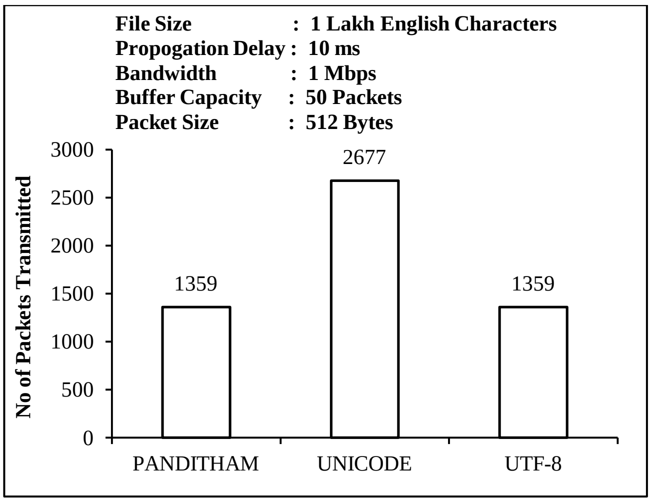

Fig.11. Packets Transmitted with Bandwidth 1 Mbps

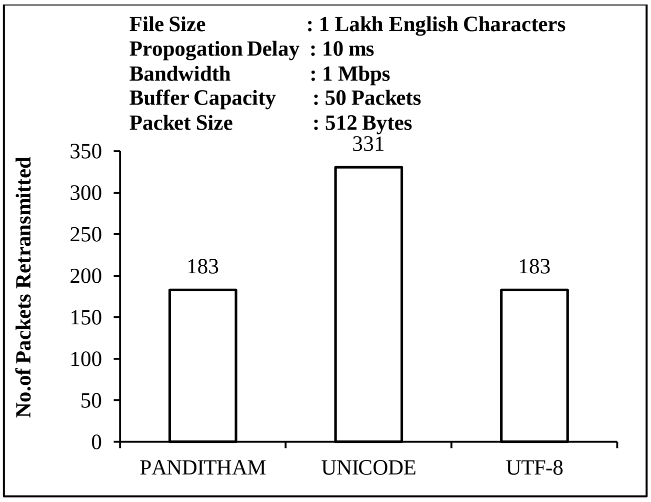

Fig.12. Packets Retransmitted with Bandwidth = 1 Mbps

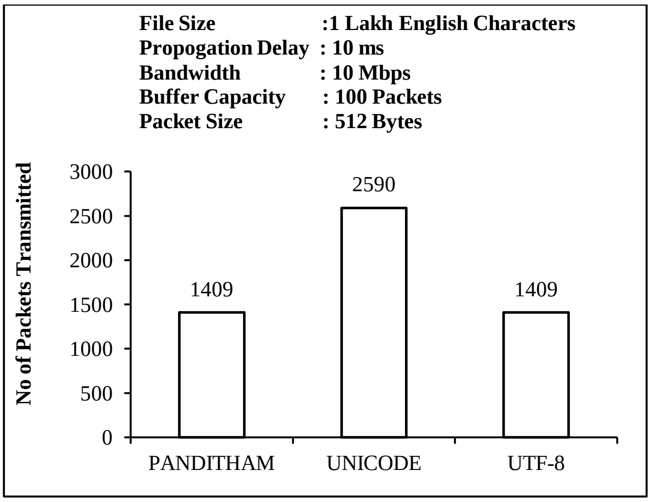

Fig.13. Packets Transmitted with Buffer Capacity = 100 packets

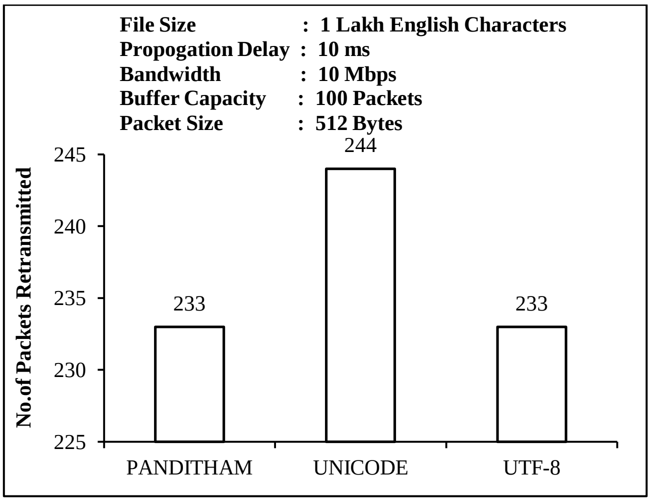

Fig.14. Packets Retransmitted with Buffer Capacity = 100 packets

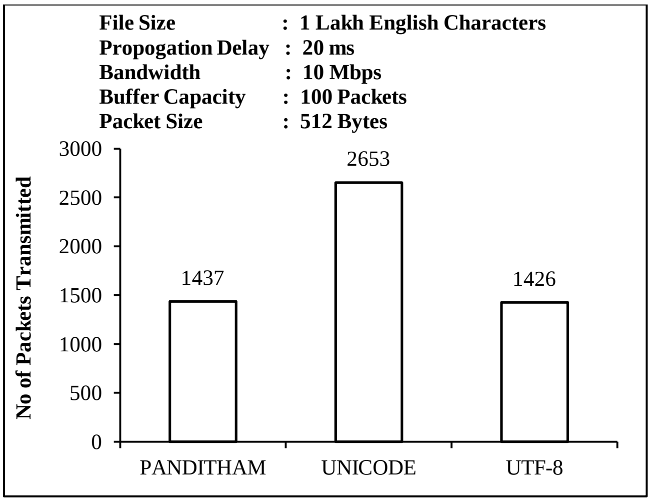

Fig.15. Packets Transmitted with Propagation Delay = 20 ms

Subsequently, the propagation delay alone is changed from 10 ms to 20 ms, and Fig.15 and Fig.16 depict the respective comparisons. From the analysis done, it is inferred that PANDITHAM and UTF-8 have taken same number of packets to transfer 1 lakh English characters compared to Unicode. When the traffic is less, more packets are transmitted in less amount of time and this will subsequently reduce the number of retransmissions and reduce the network congestion.

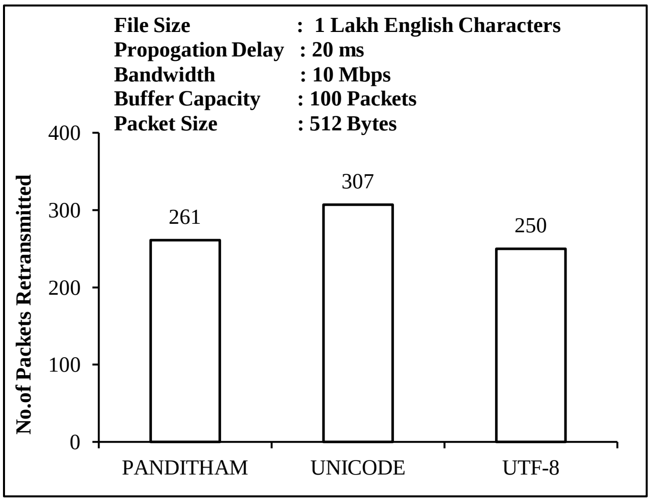

Fig.16. Packets Retransmitted with Propagation Delay = 20 ms

4.2 STUDY OF TIME NEEDED TO TRANSFER A TEXT FILE

4.2.1 Tamil Text:

When a file with one lakh Tamil Character is transmitted using the propagation delay of 10 ms and with the buffer capacity of the queue as 50 packets, then the time needed to transfer the packets with a bandwidth of 1 Mbps using PANDITHAM, UNICODE and UTF-8 are shown in Fig.17.

In the previous setup, the bandwidth alone is changed from 1 Mbps to 10 Mbps. In this context, the time taken to transfer the number of packets are depicted in Fig.18, respectively. In the next setup, the buffer capacity is changed from 50 packets to 100 packets and the corresponding comparison chart is shown in Fig.19, respectively.

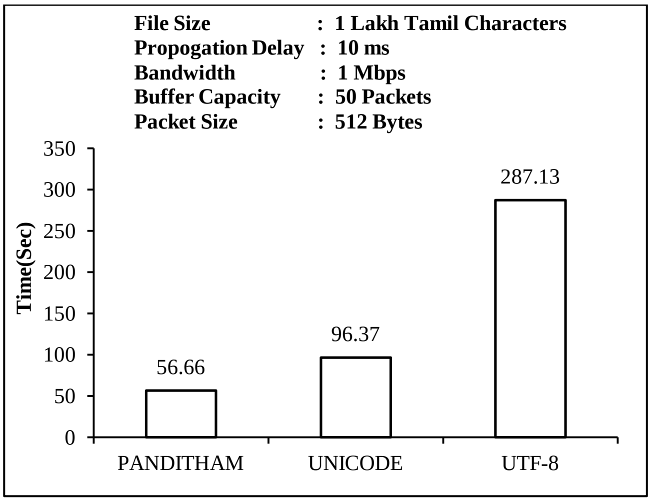

Fig.17. Time to Transfer Tamil Characters with Bandwidth=1 Mbps

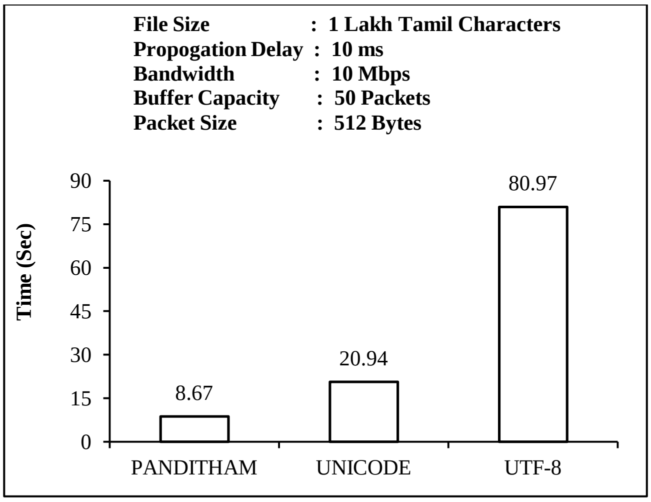

Fig.18. Time to Transfer with Bandwidth = 10 Mbps

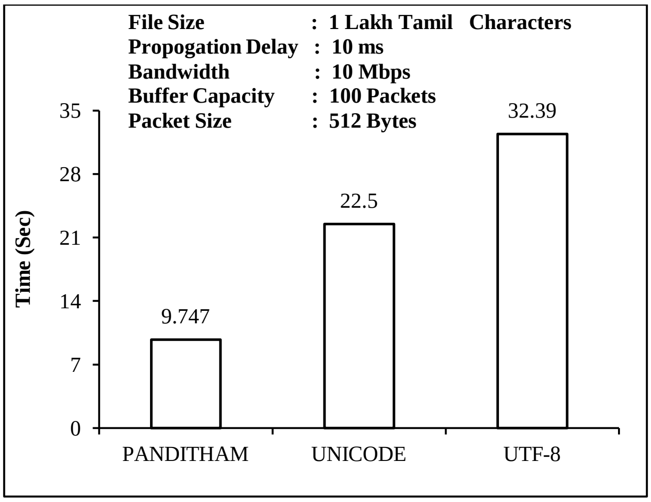

Fig.19. Time to Transfer with Buffer Capacity = 100 packets

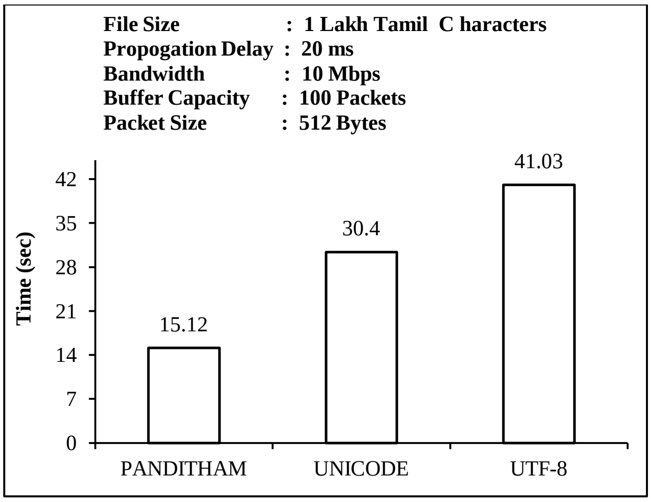

Fig.20. Time Transfer with Propagation Delay = 20 ms

Subsequently, the propagation delay alone is changed from 10 ms to 20 ms, is shown in Fig.20. The time analysis shows that PANDITHAM have taken less amount of time to transfer one lakh Tamil characters when compared with Unicode and UTF-8. So it is clear that PANDITHAM can transfer packets much faster than its counterpart Unicode and UTF-8.

4.2.2 English Text:

When a file with one lakh English Character is transmitted using the propagation delay of 10 ms and with a buffer capacity of the queue as 50 packets, then the time needed to transfer the packets, with a bandwidth of 1 Mbps using PANDITHAM, UNICODE and UTF-8 are shown in Fig.21.

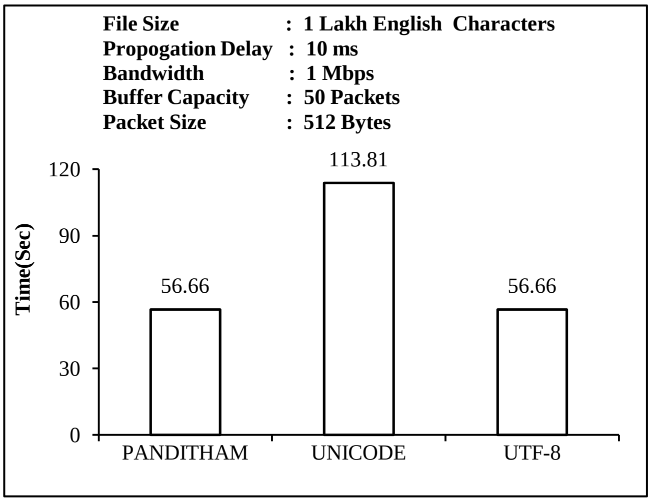

Fig.21. Time Transfer with Bandwidth = 1 Mbps

In the previous setup, the bandwidth alone is changed from 1 Mbps to 10 Mbps. In this context, the times taken to transfer the number of packets are depicted in Fig.22, respectively. In the next setup, the buffer capacity is changed from 50 packets to 100 packets and the corresponding comparison chart is shown in Fig.23, respectively. Subsequently, the Propagation delay alone is changed from 10ms to 20ms, is shown in Fig.24.

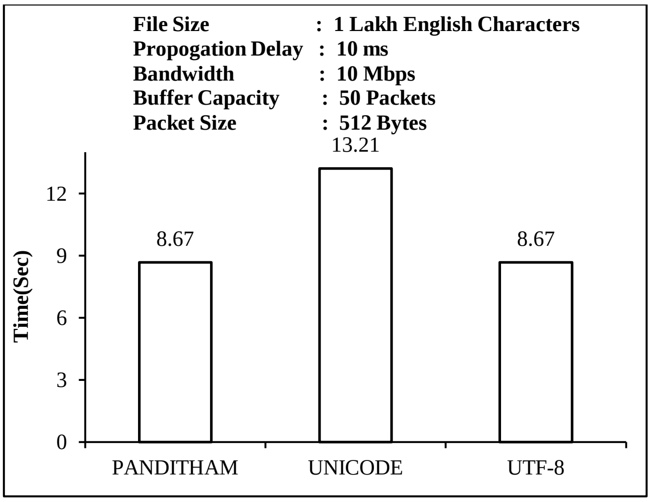

Fig.22. Time to Transfer with Bandwidth = 10 Mbps

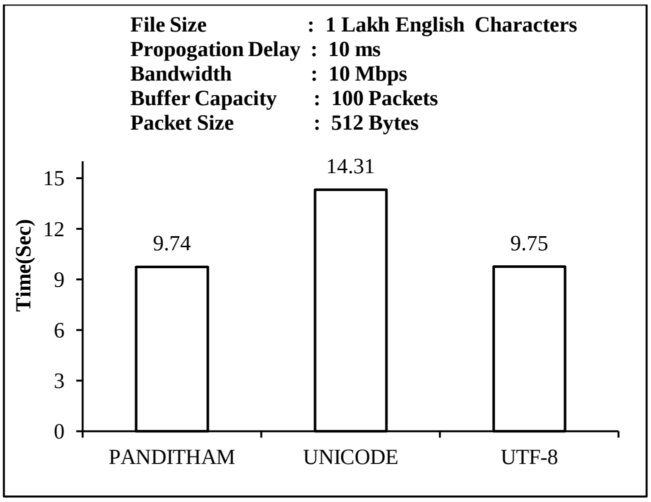

Fig.23. Time to Transfer with Buffer Capacity = 100 packets

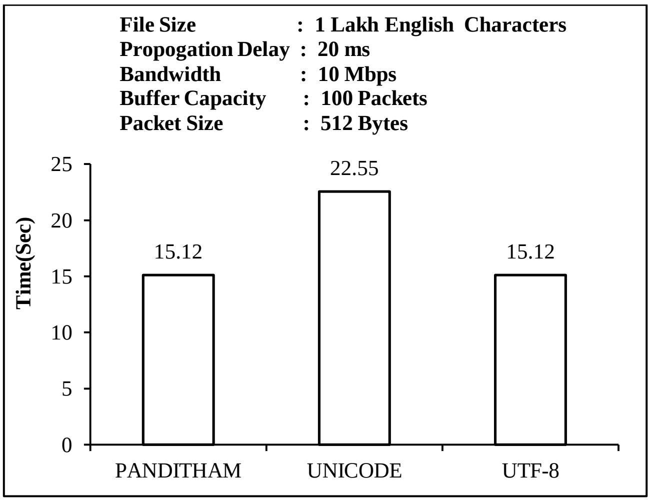

Fig.24. Time to Transfer with Propagation Delay = 20 ms

5. CONCLUSION

From the traffic analysis done, it is found that the rapidly accelerating trend of globalization of businesses and the success of e-Governance solutions require data to be stored and manipulated in many different natural / local languages. When the data is encoded using PANDITHAM as character encoding scheme for different languages more packets can be transmitted in less amount of time. The network congestion can be significantly reduced by reducing the number of retransmissions. It can be used for all other languages as well where the characters in the language have to be stored in a PANDITHAM table, and the language codes have to be standardized worldwide.

REFERENCES

[1] P. Navaneethan, R. Madheswaran, R. Balasubramaniam and R.V. Bharathidasan, "PANDITHAM: An Optimal Character-Oriented Protocol for Multilingual Computing", *Tamillnaiyam*, 2000.

[2] Yilu Zhou, Jialun Qin, Hsinchun Chen, Jay F. Nunamaker: "Multilingual Web Retriveal: An Experiment on a Multilingual Business Intelligence Portal", *Proceedings of the 38th Hawaii International Conference on System Sciences*-2005.

[3] A.Kumaran and R.Haritsa, "On Database Support for Multilingual Environments", *Thirteenth International Work Shop on Research Issues in Data Engineering: Multi-lingual Information Management, IEEE Computer Society*, 2003.

[4] ISO. ISO/IEC 8859 *Information Processing-8-bit Single Byte Graphic Coded Character Sets*, ISO/IEC 8859-15:1999.

[5] Global Reach, "*Global Internet Statistics*", available at - http://www.glreach.com/globstats, 2002.

[6] Venkat V.S.S.Sastry and Lakshmi Sastry, "*Teach Yourself Tcl/Tk in 24 hours*", Sams Publications, 1999.

[7] Tim Lindholm and Frank Yellin, "*Java Virtual Machine Specification*" 2nd Edition, Addison-Wesley, 1999.

[8] www.unicode.org

[9] www.cdac.in

[10] www.wikipedia.com

[11] www.isi.edu