Performance of TCP/IP/ UDP adaptive header compression algorithm for wireless network

Prof.N..Penchalaiah
Department of Computer Science Engineering
ASCET
Gudur, India

Dr.R.Seshadri
Prof & Director of university computer center
S.V.University
Tirupati, India

***Abstract*: The convergence of mobile technologies will characterize the future telecommunication systems based on TCP/IP protocols. These Protocols can be used to build both wired and wireless parts on same platform. But TCP/IP headers overheads are large, so it utilizes more bandwidth even small services, whose payload is small. So it definitely need header compression to save bandwidth, as we know bandwidth is the most expensive and scared resource in wireless systems.There are many approaches used to compress the headers overloads. Most commonly used approaches are, an adaptive robust TCP/IP header compression algorithm for 3G wireless networks and an adaptive robust header compression algorithm based on UDP-RTS/CTS handshake for real-time streams in wireless networks such a 3G platforms. But these approaches are not efficient in adaptive nature. So In this paper we are proposing one approach to solve the problem efficiently even in adaptive nature. The aim of this algorithm is to adjust the dimension of Variable Sliding Window (VSW) in W-LSB encoding with the accurate estimation of wireless channel state to achieve the good balance of compression ratio and error-resistant robustness for the adaptive use in wireless link. We present simulation results that demonstrate the effectiveness of this adaptive algorithm over wireless link and comparative study of existing approaches**

Keywords : TCP/IP, UDP, Wireless Network, Adaptive, W-LSB algorithm

I. INTRODUCTION

The third generation (3G) mobile communication system is expected to develop on a network platform totally based on Internet protocols[1], Because, the future telecommunication systems will be surely characterized by the convergence of mobile technologies and Internet protocols, achieved through a network platform totally based on TCP/IP protocols ("all-IP network")[2].That includes cellular links, whose spectrum is the most scarce resource of the whole wireless system. However, because of the encapsulation process of each layer of the hierarchical TCP/IP[4,5] architecture, a substantial part of the radio bandwidth will be wasted for the transmission of control information (header) that do not have any specific function for the management of radio channel itself.

Moreover, for applications characterized by a small payload, a large header can introduce a significant extra delay that makes unrealizable the service on the TCP/IP platform. For wireless system, these problems are becoming of increasing importance for two principal reasons: the limited bandwidth available and the wide use of new protocols (as IPv6[6] and Mobile IP[7]) with variegated and more complex headers. For example an application based on a TCP/IPv4 protocol suite, whose header length is 40 bytes, increases to 60 bytes with a TCP/IPv6 stack, and with mobile IPv6, the header grows to 100 bytes.

To utilize the limited bandwidth efficiently, TCP header compression was introduced to reduce the large header overhead by sending only the changing fields in the packet header. The error-proneness (the bit error rate as high as 10e-3, even 10e-2) and the large delays (the round trip time as high as 100-200ms) of cellular links make header compression as defined in[8,9] perform less than well. A viable header compression scheme for usage in cellular systems must produce very small headers to enable efficient usage of the scarce spectrum while still being robust to the error patterns.

On the other hand, all existing header compression schemes do not take wireless channel state into consideration when designed. They just adopt variations of encoding methods and repair mechanisms to minimize the number of lost packets when error happens over wireless links. If the current channel state can be predicted, the header compression scheme may take some actions and mechanisms in advance of compressing headers. Obviously, this type of header compression scheme may suit much better to be used over wireless links.

In this paper, a new adaptive robust TCP/IP header compression algorithm for 3G wireless networks and an adaptive robust header compression algorithm based on RTS/CTS handshake for real-time streams in wireless networks such as 3G platforms is proposed and analyzed. By adjusting the dimension of Variable Sliding Window (VSW) of Window-based Least Significant Bits (WLSB) encoding in header compressor with the accurate estimation of wireless channel state, this adaptive algorithm can not only compress the TCP/IP/UDP headers robustly, but also achieve rather high efficiency. We present experiment results that demonstrate the effectiveness of this algorithm over simulated wireless links

II. PROBLEM DESCRIPTION AND EVALUATION

A. TCP/IP Header Compression: State Of Art

Header compression is possible for the presence of significant redundancy among header fields, both within the same packet header but, in particular, among consecutive packets belonging to the same packet stream Thus, by sending static fields information only initially, and utilizing dependencies and predictability for other fields, the header size can be significantly reduced for most packets. Generally, as illustrated in Fig.1, header compression Algorithms maintain, at both the compressor and the decompressor sides, a shared state called context including relevant information which compression and decompression are done relative to. Normally, the decompressor will keep synchronized with the compressor, thus it can decompress the compressed packet header correctly. But when packets are lost over the link, the decompressor context will be brought out of sync with compressor context. and decompressing of subsequent headers will fail. This effect, so-called error propagation, which will last until the contexts are brought into synchronization by some means, will degrade the performance of header compression, especially on relative high BER link, such as wireless channel.

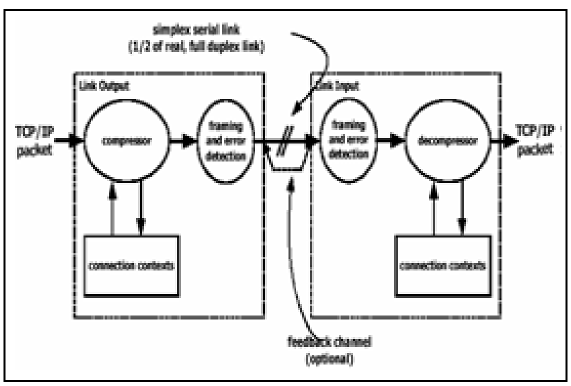

Figure 1. A general header compression scheme

The original TCP/IP header compression scheme is proposed by Van Jacobson (VJHC) in [7]. In most cases, VJHC can compress the 40 octets full TCP/IPv4 header to only 4 octets and improves the TCP performance significantly on low-speed links. However, due to the differential encoding resulting in frequent context out of sync once a delta lost on the Link, its performance on a high BER wireless link has a breakdown[8,9]. successively, a number of other protocols have been written. Noteworthy is the algorithm proposed by Degermark (IPHC) in [9], Who suggest two simply mechanisms, twice algorithm and the explicit header request mechanism to improve the header compression synchronization robustness.

However, the twice algorithm only works well under some assumptions. The explicit request mechanism needs a feedback channel and its performance is impacted significantly by the Link round trip time. On loss links with long round trip times, such as most wireless links, IPHC does not perform well[9,10].A viable TCP/IP header compression scheme for usage in Wireless systems is needed which must produce very small headers while still being robust to the error patterns. ROHC (Robust Header Compression)[12], the header compression scheme that the homonymous workgroup of the IETF (Internet Engineering Task Force) is developing, satisfies all the requirements for a valid header compression scheme that is destined to a wireless link. However ROHC is not able to manage, efficiently, a stream based on TCP. ROHC+[12] Proposed a specific header compression profile for TCP streams within the ROHC platform. This scheme is based on a distinct management of data and acknowledgement streams associated to the TCP stream. To reduce error propagation, ROHC+ adopts a robust encoding technique (W-LSB) and the synergy between three repair mechanisms for damaged compressed packets.

B. Adaptive Roburst TCP/IP Header Compression:

All these previous header compression schemes do not take wireless channel state into consideration when designed. They just adopt variations of encoding methods and repair mechanisms to minimize the number of lost packets when error happens over wireless links. If the current channel state can be predicted, the header compression scheme may take some actions and mechanisms in advance of compressing headers. Obviously, this type of header compression scheme may suit much better to be used over wireless links.

Based on this idea, we propose an adaptive robust TCP/IP header compression algorithm for 3G wireless networks. The block diagram of this adaptive algorithm is illustrated in Fig. 2.In fact, in our realization, the function of channel estimator is included in header decompressor.

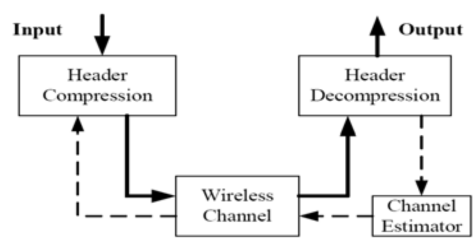

Figure 2. Block diagram of the adaptive algorithm

In this adaptive algorithm, header compressor adopts a robust encoding technique which is called window-based least significant bits (W-LSB) encoding algorithm. We propose to Use the accurate estimation of wireless channel state to control the size of Variable Sliding Window (VSW) of W-LSB encoding in order to achieve a good balance between compress ratio and robustness. This adaptive algorithm may be adaptable to the burst error and delay characteristics of wireless links over which it is used. Before introducing this algorithm it is important to brief the functionalities of W-LSB encoding that are most related to our Work.

C. Adaptive Robust TCP/IP Header Compression Algorithm

Many different variations of methods to estimate the wireless channel state have been written in the literature. In our proposed adaptive robust TCP/IP header compression algorithm, we adopt the method used in TCP-ELSA to predict channel state. The state of link is defined in terms of the ratio of its temporary loss rate to its long-term average loss rate as depicted in Fig.3 instead of defining the state of a link according to the absolute value of the loss rate. The solid line in Fig.3 represents the long-term average loss rate while the dashed line represents the short-term temporary loss rate. If a link's temporary loss rate is higher than its average loss rate, the link is considered to be in a bad state; otherwise, it is considered to be in a good state.

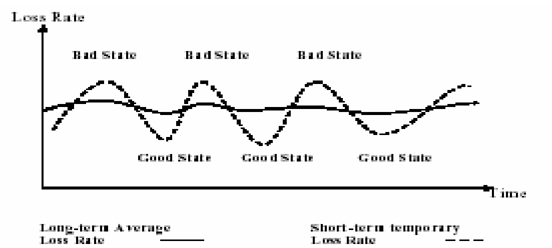

Figure 3. Good state and bad state

Our adaptive algorithm estimates the link state by maintaining two parameters: the average packet loss rate (ALR) and the temporary packet loss rate (TLR), using a weighted moving average algorithm. By assigning different parameters to the weighted moving average, ALR can represents the long term quality of the wireless link and TLR can be set to be more sensitive to transient packet loss rate changes and indicate the current packet loss rate. According to this information, our adaptive algorithm adjusts the dimension of VSW in header compressor, which is the key parameter for compression performance. When VSW is small the scheme has high compression ratio but it cannot avoid many packet losses. Instead, when VSW is large the scheme has strong robustness but weak compression ratio and weak bandwidth efficiency. The header decompressor determines the state of wireless link according to the values of ALR and TLR, and notifies the compressor the estimation result by sending a specially defined type of feedback packet. Every time receiving such type of feedback packet, the compressor adjusts the dimension of VSW accordingly. When TLR is larger than ALR for a certain threshold, which signals that the wireless link is in a temporarily bad state, the compressor increases the dimension of VSW. In this state, our algorithm can tolerate more packets lost over the wireless link by increasing the number of reference base residing in VSW. After several rounds, TLR gets lower than ALR for a certain threshold, so the wireless link is in normal good state. In this state, the compressor decreases the dimension of VSW accordingly to provide greater compression ratio. These thresholds can be set to be different values depending on the specific system and radio channel.

D. RTP/UDP/IP Header Compression: State Of Art

Header compression is possible for the presence of a significant redundancy among header fields, both within the same packet header but in particular, among consecutive packets belonging to the same packet stream. Thus, by sending static fields information only initially, and utilizing dependencies and predictability for other fields, the header size can be reduced significantly for most packets. Generally, header compression algorithms maintain, at both the compressor and the decompressor sides, a shared state called context including relevant information which compression and decompression are done relative to. Normally, the decompressor will keep synchronized with the compressor, thus it can decompress the compressed packet header correctly. But when packets are lost over the link, the decompressor context will be brought out of sync with compressor context and decompressing of subsequent headers will fail. This effect, so-called error propagation, which will last until the contexts are brought into synchronization by some means will degrade the performance of header compression, especially on relative high BER links, such as wireless channels.CRTP is suitable for compressing RTP/UDP/IP headers over low speed serial links, which can compress the 40 or 60 octets RTP/UDP/IP headers to 2-4 octets. On lossy links with long round trip times, such as most cellular links, CRTP does not perform well . A viable header compression scheme for usage in cellular systems must produce very small headers to enable efficient usage of the scarce spectrum while still being robust to the error patterns.

Many schemes have been proposed to make header compression suit to be used in wireless environment. ACE (Adaptive Header ComprEssion) and ROCCO (Robust Checksum-based header Compression) are two significant among these schemes. ACE introduces an innovative W-LSB encoding algorithm for changing fields that uses many reference values contained in VSW (Variable Sliding Window) in the Compressor, while the decompressor can use any value of these references to decompress correctly. ROCCO uses a CRC included in compressed headers to verify correct reconstruction in the decompressor and to avoid error propagation. The IETF Robust Header Compression Working Group (ROHC WG) is studying new header compression schemes that are having good performances over wireless links of 3G cellular systems. However, the schemes should also be applicable to other future link technologies with great loss and long round-trip times. Ideally, it should be possible to compress over unidirectional links. A new robust header compression scheme (ROHC) which uses and combines all the techniques developed by ACE and ROCCO has been just proposed which aims at providing a compression scheme that has high compression efficiency and high robustness when used in wireless links.

E. Adaptive Robust Header Compression Algorithm Based on UDP-RTS/CTS Handshake

Many different variations of methods to estimate the wireless channel state have been proposed in the literature. In this paper, we adopt the method based on RTS/CTS handshake mechanism used in to predict channel state. The header compressor sends an RTS (ready-to-send message) to the decompressor, indicating that it has n packets with compressed headers to send. The decompressor may not receive the RTS correctly due to errors. If it does, it replies with a CTS (clear-to-send reply) accepting the transmission. The compressor may also not receive the CTS reply, because, for example, the link is currently in a bad state. In this last case too he will perceive an RTS/CTS failure. If the compressor fails with the RTS/CTS transmission, it persists, up to a specific maximum number of times.

We assume that the probabilities of unsuccessful RTS/CTS completion are incorporated in the two-state (good state, and bad or faded state) Markov model that we use to model the link between the compressor and decompressor. The RTS/CTS message exchange has been used for the purpose of probing the state of the channel. However, RTS/CTS use imposes an overhead to the channel throughput. Since the size of RTS and CTS packet (in our simulation, they are both set to be 1 byte) is much smaller than that of an actual packet, the throughput loss introduced by RTS/CTS is much less. What's more, when the value of n is set to be larger than 1, the overhead of RTS/CTS becomes even less. But in this case, the accuracy of channel state estimation decreases. Many studies in literature such as 113, 141 have established that finite state Markov models can be effectively used to characterize the error behavior of wireless links.

We adopt a variation of this approach in our work we model the actual links as two-state Markov models. In header compressor, the dimension of VSW is adjusted according to the current state of link. When the link is in good state, the compressor should decrease the dimension of VSW to get a bigger compression ratio. Otherwise, when the link is in bad state, the dimension of VSW should be increased to get a better robustness to resist the non-trivial packet loss in wireless link. Ideally we would like the compressor to have exact knowledge of the state each link is in. But this in reality never happens. So in our simulations the compressor keeps its estimate of how good it thinks the link. The goodness of the link is expressed through parameter g. The compressor updates this parameter whenever n packets headers are to be compressed according to the current state of channel.

The parameter g is allowed to take a number of integer values. The intuition is, that the better we think the link to the decompressor is (the bigger the g). the more we should persist when we try to transmit to the decompressor, because we indeed have better chances to succeed. What's more, the smaller we should set the dimension of VSW, because the probability of packet loss in the link is much small. On the other hand, if we think that the link is really had, we should not lose much time on it and maintain a VSW whose dimension is big enough to assure the decompressor would not come into the inconsistency with the compressor in case of there are many packets lost in the link. In our simulations g is translated to the max number of RTS attempts allowed: if CTS is not successfully received even after g tries at sending RTS, then g is decreased exponentially. If we succeed with less than g tries, g is increased (up to a max value) inversely proportional to the number of attempts. Allowing g to take more than two values increases the flexibility of the channel-state-dependent

module. Multiple values of g permit the compressor to be conservative and take into account its previous knowledge of the link state.

In reality good state is not error-free or bad state always destructive, so in order not to have purposeless oscillations; one would like to gradually build up the link state estimate based on more than one link probe. The performance depends on the specific functions used to update g, and is a choice for each system since no generic solution could possibly be optimum for all the different optimizations desired. Another benefit of a multivalued g is that in our model we have assumed a two-state only Markov-model for the wireless link to the header decompressor. However, depending on the specific system and radio channel, a multiple state Markov model may be better. In such a case the "estimate.

F. W-LSB Encoding (TCP/IP):

Previous header compression schemes VJHC and IPHC do not perform well on wireless links, which are characterized by high bit error rate and transmission delay because the decompressor is required to use always the same reference base value that was used by the compressor to in order to decompress the encoded value correctly. However, the packet including this reference base value could be lost on the channel because of bad channel conditions. The innovative Window based Least Significant Bits encoding (W-LSB encoding) does not impose this restriction because it tries to encode a value based on a group of reference base values included by the sliding window (VSW). The decompressor can decompress the encoded value correctly once any value in VSW can be delivered successfully. By using W-LSB encoding, the compressor would not come into the inconsistency with the decompressor unless all values in the VSW are lost.

W-LSB encoding is used for header fields whose values are always subject to small changes among consecutive packets (changing fields). With W-LSB encoding, the k least significant bits of the field value are transmitted instead of the original field value because the Most Significant Bits

(MSB) remain relatively constant during many sessions. The value of k is calculated in the compressor using N reference values included in the VSW: for each reference value *v_ref*, *k_ref* is determined so that the value to compress belongs to the interpretation interval *f(v_ref ,k_ref).*

***f* (*v _ref* ,*k _ref*) = [*v _ref* +*p*, *v _ref* +2^ (*k _ref* -1) -*p*].**

Where *p* is an integer.

```
<-- -interpretation interval (size is 2^k)--- >
|----------+-----------------------------|
v_ref - p v_ref        v  _ref +(2^(k_ref-1) –p
```

For any value k_ref, the k_ref least significant bits will uniquely identify a value in f(v_ref, k_ref). The parameter p is introduced so that the interpretation interval can be shifted with respect to v_ref. Following, right value of k is the maximum among the N values of k_ref calculated above. After receiving k bits, the decompressor derives the original value using a previously received value as reference.

It should be clear how the dimension of VSW influences k and therefore the compression ratio too as k is typically a monotonous increasing function of the number of reference values N, i.e. the dimension of VSW. When N is small the scheme gives high compression ratio but losses of packets are not allowed on the channel. Otherwise, when VSW is large the scheme has strong robustness but low compression ratio and low bandwidth efficiency. When the compressor knows what values in VSW have been received by the decompressor, VSW can be shrunk to obtain a rather high compression ratio. To shrink the VSW, the compressor needs some means to get feedbacks that indicate what value has been received by the decompressor.

Considering the features of W-LSB encoding, the header compression mechanism we are going to propose introduces an innovative algorithm estimating optimum VSW dimension in wireless links: the channel state based robust header compression.

G. W-LSB Encoding (UDP-RTS/CTS):

CRTP does not perform well on wireless links, which are characterized by high bit error rate and transmission delay because the decompressor is required to use always the same reference base value that was used by the compressor to in order to decompress the encoded value correctly. However, the packet including this reference base value could be lost on the channel because of bad channel conditions. The innovative Window-based Least Significant Bit encoding (W-LSB encoding) does not impose this restriction because it tries to encode a value based on a group of reference base values included by the sliding window (VSW). The decompressor can decompress the encoded value correctly once any value in its referenced VSW can be delivered successfully. By using WLSB encoding, the compressor would not come into the inconsistency with the decompressor unless all values in the VSW are lost. W-LSB encoding is used for header fields whose values are always subject to small changes among consecutive packets (changing fields). With W-LSB encoding, the k least significant bits of the field value are transmitted instead of the original field value because the Most Significant Bits (MSB) remain relatively constant during many sessions. The value of k is calculated in the compressor using N reference values included in the VSW: for each reference value v-ref; h-ref is determined so that the value to compress belong to the interpretation interval f(v-ref, k-ref).

f(v-ref , k-ref) = ***[v- ref - p, v-ref – 2^(k*** - ref - **1) -** ***p]***
Where P is an integer

```
<--interpretation interval (size is 2^k)--->

|-------------+------------------------------------|

v-ref- p  v-ref        v-ref +(2^(k_ref-1) – p
```

For any value k-ref; the k-ref least significant bits will uniquely identify a value in f (v-ref, k-ref). The parameter p is introduced so that the interpretation interval can be shifted with respect to v-ref: Following, right value of k is the maximum among the N values of k-recalculated above. After receiving k hits, the decompressor derives the original value using a previously received value as reference. It should be clear how the dimension of VSW influences k and therefore the compression ratio too as k is typically a monotonous increasing function of the number of reference values N, i.e. the dimension of VSW. When N is small the scheme gives high compression ratio hut losses of packets are not allowed on the channel. Otherwise, when VSW is large the scheme has strong robustness but low compression ratio and low bandwidth efficiency. When the compressor knows what values in VSW have been received by the decompressor, VSW can be shrunk to obtain a rather high compression ratio. To shrink the VSW, the compressor needs some means to get feedbacks that indicate what value has been received by the decompressor. Considering the features of W-LSB encoding, the header compression algorithm we are going to propose introduces an innovative algorithm estimating optimum VSW dimension in wireless links: the adaptive robust header compression based on RTS/CTS handshake.

III. COMPARISIONS BETWEEN TCP/IP AND UDP HEADERS

TABLE 1. HEADER COMPRESSION (WITHOUT W-LSB ENCODING)

Protocol Name	TCP/IP		UDP	
Octets to be compressed	Before	After	Before	After
	40	4	40-60	2-4
Band width	limited		limited	
Packets loss	high		high	
performance	Not well		Not well	

TABLE 2. ADAPTIVE HEADER COMPRESSION (WITH W-LSB ENCODING)

Protocol Name	TCP/IP	UDP
Octets to be compressed	High	High
Band width	High	High
Packets loss	Low	Low
performance	Well	well

IV. CONCLUSION

This paper proposes a new adaptive robust TCP/IP header compression algorithm and a new adaptive robust header compression algorithm based on RTS/CTS handshake for real time streams in wireless networks such as 3G platforms. Both the algorithm adjusts the dimension of Variable Sliding Window (VSW) in W-LSB encoding with the accurate estimation of wireless channel state to achieve the good balance of compression ratio and error-resistant robustness for the adaptive use in wireless link. With applying of these algorithms, we implement a new header compression scheme: AROHC, CRTP+ and ROHC+.

REFERENCES

[1] 3GPP TR 23.821 v.1.0.1 (2000-07), Architecture principle for Release 2000, July 2000.

[2] Yichuan Wu Sun Limin Jian Ma "Adaptive robust TCP/IP header compression algorithm for 3G wireless networks" IEEE WCNC-2004,vol.2, pg.1046 – 1050.

[3] 3GPP TR 23.922 v.1.0.0 (1999-07), Architecture for an All IP Network, July 1999.

[4] W. R. Stevens, "TCP/IP Illustrated Volume 1 – The Protocols," Addison-Wesley, 1994.

[5] J. Postel., "Internet protocol," Request For Comments RFC791. DARPA September 1981.

[6] S. Deering and R. Hinden, "Internet protocol, version 6(IPv6)," Request For Comments RFC2460.IETF, December 1998.

[7] K. Svanbro, H. Hmu, L. E. Jonssan and M. Deger-mark, "Wireless real-time Ip servlca enabled by header compression," Vehicular Technology Con-ference Proceedings, 2000.

[8] C.Fragouli, VSivaraman and MSrivastava, 'Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queuing with Channel-State Dependent Packet Scheduling," IEEE infocom 1998.

[9] F. Swarts, and H Ferreira, "Markov characterization of digital fading mobile VHF channels," IEEE Trans. on Vehicular Technology, vol. 43, no. 4, pp. 977-985, Nov. 1994.

[10] H. Wang, and N. Moayeri, "Finite State Markov Channel - a useful model for radio communications," IEEE Trans. on Vehicular Technology, vol.44,no. I,pp. 163.171, Feb. 1995.

[11] http: llinfo.iet.unid.itl-luieiiir, dummmet

[I2] M.Mouly and M.B.Pautet, "me GSM system for mobile

[I3] F.Muratore, "UWZS - Mobile Communicatioon for the Furute, " Wiley,

[14] 802.11 Working Gump, http:llgrouper.ieee. orgigroupsi80211 li, May Eomm unications," Cell & Svs, 1992. 2000. 2001.

Prof.N.Penchalaiah Research Scholar in SV University, Tirupati and Working as Professor in CSE Dept,ASCET,Gudur. He was completed his M.Tech in Sathyabama University in 2006. He has 10 years of teaching experience. He guided PG & UG Projects. He published two papers in National and one in International Conferences.

Dr.R.Seshadri working as Professor & Director of University Computer Centre, Sri Venkateswara University, Tirupati. He was completed his PhD in S.V.University in 1998 in the field of "Simulation Modeling & Compression of E.C.G. Data Signals (Data compression Techniques) Electronics & Communication Engg.". He has richest of knowledge in Research field, he is guiding 10 Ph.D in Fulltime as well as Part time. He has vast experience in teaching of 26 years. He published in 10 national and international conferences and 8 papers published different Journals.