SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING

A Constituent College of

JSS SCIENCE & TECHNOLOGY UNIVERSITY

TRANSMISSION CONTROL PROTOCOL IN SENSOR SYSTEM

Mini project final report submitted in partial fulfillment of curriculum prescribed for the Computer Networks (20CS520) course for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PROJECT REPORT

Submitted by

S.No.	NAME	ROLLNO.	USN
1	Adithya Deepthi Kumar	28	01JST21CS005
2	E Shreyas Herale	35	01JST21CS037
3	Eshwar J	8	01JCE21CS033
4	Harsha N P	10	01JCE21CS038

D-SECTION

Under the Guidance of

Dr.S Manjula

Assistant Professor

Dept.of CS & E,

SJCE, JSSSTU, Mysuru

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

December 2023

SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING
A Constituent College of
JSS SCIENCE & TECHNOLOGY UNIVERSITY

CERTIFICATE

This is to certify that the work entitled **"TRANSMISSION CONTROL PROTOCOL IN SENSOR SYSTEM "** is a bonafied work carried out by **Adithya Deepthi Kumar, E Shreyas Herale, Eshwar J, Harsha N P** in partial fulfillment of the award of the degree of **Bachelor of Engineering in Computer Science and Engineering of JSS Science and Technology University, Mysuru during the year 2023**. It is certified that all corrections / suggestions indicated during CIE have been incorporated in the report. The mini project final phase report has been approved as it satisfies the academic requirements in respect of mini project work for event 2 prescribed for the Computer Networks (20CS520) course.

Course in Charge and Guide

Dr.S Manjula
Assistant Professor
Dept.of CS & E,
SJCE, JSSSTU, Mysuru

Place: Mysuru **Date : 27/12/2023**

ABSTRACT

Communication protocols play a pivotal role in establishing reliable and efficient data exchange in various network environments. This report delves into the intricate details of two essential protocols—Transmission Control Protocol (TCP) in Sensor Networks. The integration of these protocols is critical for ensuring seamless communication in diverse technological applications.

TABLE OF CONTENTS

INTRODUCTION

This report presents a detailed examination of two crucial communication protocols, namely Transmission Control Protocol (TCP) in Sensor Networks. The integration of these protocols is pivotal in ensuring efficient and reliable communication in diverse technological domains.

The first section focuses on the Transport Control Protocol, particularly its adaptation and implementation in sensor networks. Sensor networks, characterized by resource-constrained devices and dynamic topologies, present unique challenges for traditional TCP. The report delves into the modifications and optimizations required to enhance TCP's performance in such environments, addressing issues like energy efficiency, latency, and adaptability to network changes.

Furthermore, the report investigates the interplay between TCP in sensor networks examining the challenges and opportunities in integrating these protocols to enable efficient data transfer and communication in sensor-based applications. The synergy between these protocols is crucial for the successful deployment of sensor networks in various domains, including environmental monitoring, healthcare, and industrial automation.

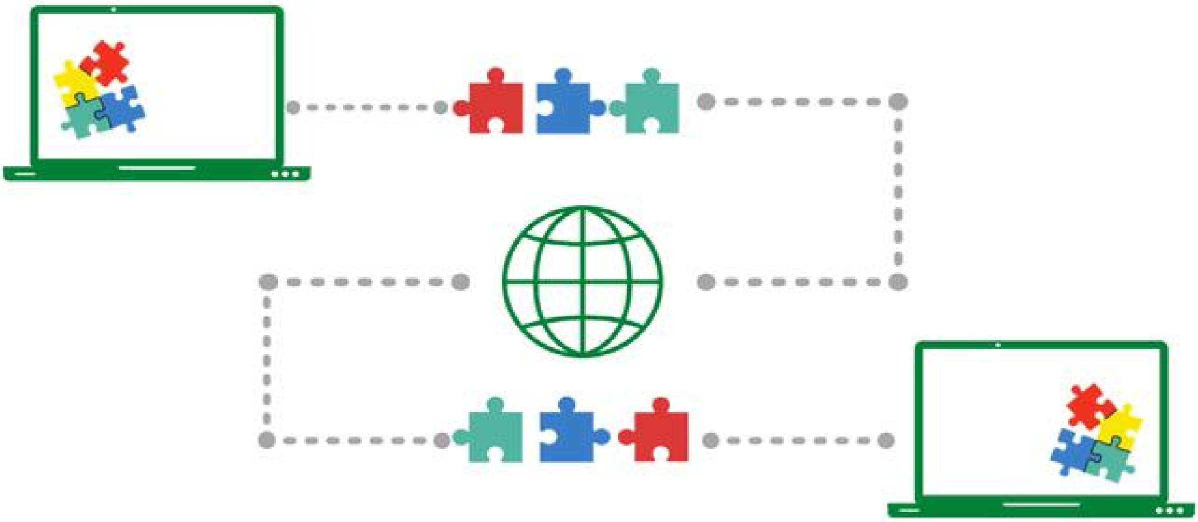
In conclusion, this report sheds light on the intricacies of Transport Control Protocol in sensor networks , emphasizing their significance in the broader landscape of communication protocols. The insights provided serve as a valuable resource for researchers, engineers, and practitioners working on the design, implementation, and optimization of communication systems in resource-constrained and dynamic network environments.

TRANSMISSION CONTROL PROTOCOL

WORKING OF TCP

To make sure that each message reaches its target location intact, the TCP/IP model breaks down the data into small bundles and afterward reassembles the bundles into the original message on the opposite end. Sending the information in little bundles of information makes it simpler to maintain efficiency as opposed to sending everything in one go.

After a particular message is broken down into bundles, these bundles may travel along multiple routes if one route is jammed but the destination remains the same.



Features of TCP/IP

For example, When a user requests a web page on the internet, somewhere in the world, the server processes that request and sends back an HTML Page to that user. The server makes use of a protocol called the HTTP Protocol. The HTTP then requests the TCP layer to set the required connection and send the HTML file.
Now, the TCP breaks the data into small packets and forwards it toward the Internet Protocol (IP) layer. The packets are then sent to the destination through different routes.

The TCP layer in the user's system waits for the transmission to get finished and acknowledges once all packets have been received.

Features of TCP/IP

Some of the most prominent features of Transmission control protocol are

1. Segment Numbering System

TCP keeps track of the segments being transmitted or received by assigning numbers to each and every single one of them.
A specific Byte Number is assigned to data bytes that are to be transferred while segments are assigned sequence numbers.
Acknowledgment Numbers are assigned to received segments.

2. Connection Oriented

It means sender and receiver are connected to each other till the completion of the process.
The order of the data is maintained i.e. order remains same before and after transmission.

3. Full Duplex

In TCP data can be transmitted from receiver to the sender or vice – versa at the same time.
It increases efficiency of data flow between sender and receiver.

4. Flow Control

Flow control limits the rate at which a sender transfers data. This is done to ensure reliable delivery.
The receiver continually hints to the sender on how much data can be received (using a sliding window)

5. Error Control

TCP implements an error control mechanism for reliable data transfer
Error control is byte-oriented
Segments are checked for error detection
Error Control includes – Corrupted Segment & Lost Segment Management, Out-of-order segments, Duplicate segments, etc.

6. Congestion Control

TCP takes into account the level of congestion in the network
Congestion level is determined by the amount of data sent by a sender

Advantages

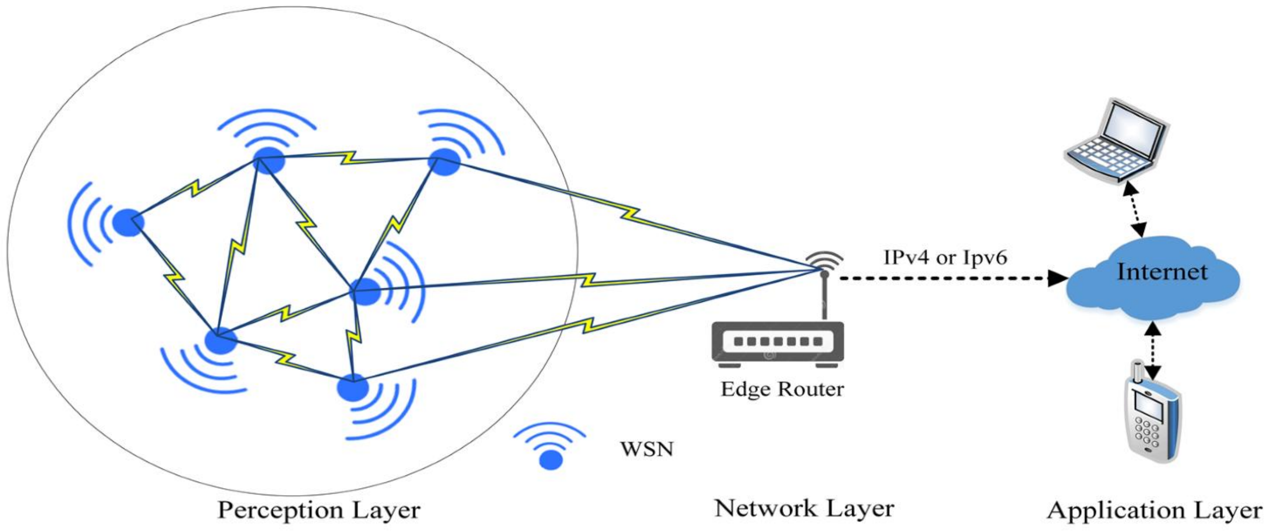
- It is a reliable protocol.
- It provides an error-checking mechanism as well as one for recovery.
- It gives flow control.
- It makes sure that the data reaches the proper destination in the exact order that it was sent.
- Open Protocol, not owned by any organization or individual.
- It assigns an IP address to each computer on the network and a domain name to each site thus making each device site to be distinguishable over the network.

Disadvantages

- TCP is made for Wide Area Networks, thus its size can become an issue for small networks with low resources.
- TCP runs several layers so it can slow down the speed of the network.
- It is not generic in nature. Meaning, it cannot represent any protocol stack other than the TCP/IP suite. E.g., it cannot work with a Bluetooth connection.
- No modifications since their development around 30 years ago.

Transport Protocols for Wireless Sensor Networks: State-of-the-Art and Future Directions

Characteristics of wireless sensor networks, specifically dense deployment, limited processing power, and limited power supply, provide unique design challenges at the transport layer. Message transmission between sensor nodes over a wireless medium is especially expensive. Care must be taken to design an efficient transport layer protocol that combines reliable message delivery and congestion control with minimal overhead and retransmission. Sensor networks are created using low cost, low power nodes. Wireless sensors are assumed to have a finite lifetime; care must be taken to design and implement transport layer algorithms that allow maximum network lifetime. In this paper we present current and future challenges in the design of transport layers for sensor networks. Current transport layer protocols are compared based on how they implement reliable message delivery, congestion control, and energy efficiency.



Wireless sensor networks (WSNs) provide a powerful means to collect information on a wide variety of natural phenomena. WSNs typically consist of a cluster of densely deployed nodes communicating with a sink node which, in turn, communicates with the outside world. WSNs are constrained by low power, dense deployment, and limited processing power and memory. WSNs are composed of small, cheap, self-contained, and disposable sensor nodes. The unique constraints imposed by WSNs present unique challenges in the design of such networks.
The need for a transport layer to handle congestion and packet loss recovery in WSNs has been debated; the idea of a cheap, easily deployable network runs contrary to the costly, lengthy process of implementing a unique and specialized transport layer for a WSN. WSNs have advanced to the level of specialization where congestion control and reliability can be incorporated at each individual node.
Reliable data transmission in WNSs is difficult due to the following characteristics of WSNs:
•limited processing capabilities and transmission range of sensor nodes;
•close proximity to ground causes signal attenuation or channel fading which leads to asymmetric links;
•close proximity to ground and variable terrain also leads to shadowing which can effectively isolate nodes from the network;
•conservation of energy requires unused nodes and wake only when needed;
•dense deployment of sensor nodes creates significant channel contention and congestion.
The above characteristics can cause loss of data in WSNs. Fortunately, WSNs also provide unique features that can be leveraged to help mitigate losses and design energy-efficient transport layer protocols by network designers. For example,
1.When the nature of the data allows, it can be aggregated at intermediate nodes.
2.Network density, multiple paths to any given destination, and data aggregation in combination with a good choice of network layer can lessen some of the losses due to channel fading and shadowing.
3.Some amount of loss can be made acceptable by employing data aggregation at the sensor nodes.

4.Data aggregation may result in smaller packet size and consequently lower packet loss.
5.Granularity of sensing an event can be controlled.
6.Some events may require a very rough granularity.
Traditional transport layer protocols, such as TCP, are not suitable for severely resource constrained WSNs having characteristics which are different from traditional wired networks. The *objective* of this paper is to illustrate the need for a standard transport layer in WSNs, outline future challenges involved in designing a transport layer protocol that fits the unique constraints imposed by WSNs, and present current implementations of transport layers for WSNs.
The *difference* between this paper and previous papers on transport layers in WSNs is that, instead of proposing a new transport layer protocol, we discuss the issues and challenges in the design of transport layer protocols. The *contribution* of this paper is to illustrate the unique requirements of a transport layer protocols for sensor networks, and compare a number of transport layer protocols that have been proposed in the literature.

It seems like you're interested in information about the Transport Control Protocol (TCP) in sensor systems. The use of TCP in sensor systems presents unique challenges due to the constraints associated with sensor networks, such as limited power, processing capabilities, and variable communication conditions. Here's some information on this topic:

Challenges of Using TCP in Sensor Systems:

1. Low Power Consumption:
   - Sensor nodes are often powered by batteries, making energy efficiency a critical concern.
   - TCP's continuous acknowledgment mechanism may lead to increased energy consumption, affecting the overall network lifetime.

2. Limited Processing and Memory:
   - Sensor nodes have constrained processing power and memory, restricting their ability to handle complex TCP features.

3. Variable Communication Conditions:
   - Sensor nodes may operate in environments with variable signal strength, leading to challenges in maintaining reliable connections.

Solutions and Adaptations:

1. Energy-Efficient Variants:
   - Researchers have proposed energy-efficient variants of TCP, such as Low-Power Wireless Personal Area Network (6LoWPAN) and Sensor Transmission Control Protocol (STCP), which aim to minimize energy consumption.

2. Adaptive Transmission Strategies:
   - TCP modifications may include adaptive transmission strategies that consider the dynamic nature of sensor networks, adjusting parameters based on network conditions.

3. Cross-Layer Design:
   - Cross-layer design approaches integrate functionalities across different layers of the communication stack, enabling better coordination between the transport layer (TCP) and the network layer in sensor systems.

4. Data Aggregation:
   - To address the challenge of limited bandwidth and energy consumption, data aggregation techniques may be employed at the sensor nodes, reducing the amount of transmitted data.

Research Areas and Future Directions:

1. Congestion Control in Wireless Sensor Networks:
   - Investigating efficient congestion control mechanisms tailored for sensor networks to prevent network degradation under heavy traffic conditions.

2. Adaptive Protocols:
   - Developing adaptive transport layer protocols that can dynamically adjust to the changing conditions of sensor networks.

3. Security Considerations:
   - Exploring security aspects related to TCP in sensor systems, ensuring data integrity and confidentiality.

4. Standardization Efforts:
   - Collaborative efforts to establish standards for transport layer protocols in sensor systems, considering the unique characteristics and requirements of these networks.

In summary, the use of TCP in sensor systems requires careful consideration of the specific challenges posed by sensor networks. Ongoing research aims to develop and optimize transport layer protocols to meet the unique constraints while ensuring reliable and energy-efficient communication.

System Implementation

AIM:

To implement a Transport Control Protocol in sensor network through the simulator.

SOFTWARE REQUIREMENTS:

Network Simulator -2

THEORY:

The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite. It originated in the initial network implementation in which it complemented the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating by an IP network. Major Internet applications such as the World Wide Web, email, remote administration, and file transfer rely on TCP. Applications that do not require reliable data stream service may use the User Datagram Protocol (UDP), which provides a connectionless datagram service that emphasizes reduced latency over reliability.

ALGORITHM:

1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.
4. Create six nodes that forms a network numbered from 0 to 5
5. Create duplex links between the nodes and add Orientation to the nodes for setting a LAN topology
6. Setup TCP Connection between n(0) and n(4)
7. Apply FTP Traffic over TCP
8. Setup UDP Connection between n(1) and n(5)
9. Apply CBR Traffic over UDP.
10. Schedule events and run the program.

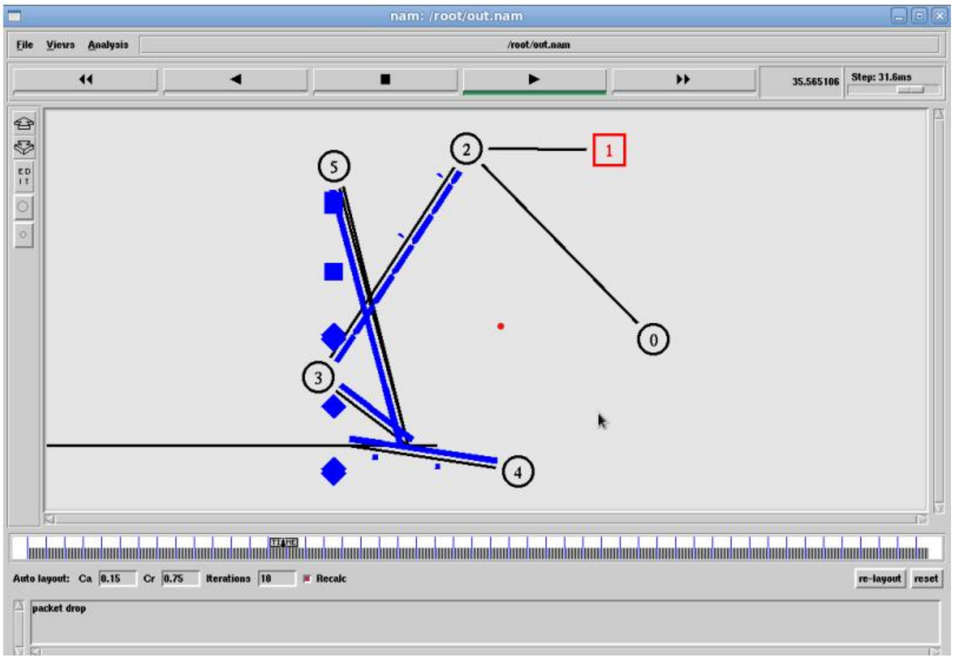
PROGRAM:

```
set ns [new Simulator]
# Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

# Open the Trace files
set file1 [open out.tr w]
set winfile [open WinFile w]
$ns trace-all $file1

# Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2

# Define a 'finish' procedure
proc finish {} {
   global ns file1 file2
   $ns flush-trace
   close $file1
   close $file2
   exec nam out.nam &
   exit 0
}

# Create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

$n1 color red
$n1 shape box

# Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
```

```
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail

set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd
Channel]

# Setup a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552

# Setup an FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp $ftp
$ftp set type_ FTP
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"

# Next procedure gets two arguments: the name of the
# TCP source node, will be called here "tcp",
# and the name of the output file.
proc plotWindow {tcpSource file} {
   global ns
   set time 0.1
   set now [$ns now]
   set cwnd [$tcpSource set cwnd_]
   set wnd [$tcpSource set window_]
   puts $file "$now $cwnd"
   $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 5 "$ns trace-annotate \"packet drop\""
```

```
# PPP
$ns at 125.0 "finish"
$ns run
```

OUTPUT:



Explanation:

The program is a script written in the TCL scripting language that uses the Network Simulator (NS-2) to simulate a simple network scenario involving six nodes, TCP connections, and FTP traffic. Here's a breakdown of the key components and actions in the script:

1. Node and Link Setup:
   - Six nodes (n0 to n5) are created using `$ns node`.
   - Colors and shapes are assigned to nodes using `$n1 color red $n1 shape box`.
   - Links between nodes are established with different characteristics (e.g., bandwidth, delay) using `$ns duplex-link` and `$ns simplex-link`.

2. LAN Configuration:
   - A LAN is created using `$ns newLan`. Nodes n3, n4, and n5 are part of this LAN.

3. TCP Connection Setup:
   - A TCP connection is set up between nodes n0 and n4 using `set tcp [new Agent/TCP/Newreno]` and `$ns attach-agent $n0 $tcp`.

- A TCP sink is created at node n4 to receive the TCP traffic.

4. FTP Application Setup:
   - An FTP application is created using `set ftp [new Application/FTP]` and attached to the TCP agent.
   - The FTP application is scheduled to start at time 1.0 seconds and stop at time 124.0 seconds using `$ns at`.

5. Window Size Plotting:
   - The script defines a procedure `plotWindow` to plot the window size of the TCP connection over time.
   - The script schedules the execution of this procedure at regular intervals using `$ns at`.

6. Trace and Nam Files:
   - Trace information is written to the "out.tr" file, which includes details of the simulation.
   - Network Animator (Nam) trace information is written to the "out.nam" file for visualization.

7. Annotation and Termination:
   - An annotation for packet drop is scheduled to be displayed at 5 seconds using `$ns at 5 "$ns trace-annotate \"packet drop\""`.
   - The simulation is set to finish at 125 seconds using `$ns at 125.0 "finish"`.

8. Execution:
   - The script initiates the simulation using `$ns run`.

CONCLUSION

The Transmission Control Protocol (TCP) plays a crucial role in network sensor systems, offering reliable and ordered delivery of data packets across networks. Its significance lies in its ability to facilitate communication integrity, ensuring that data transmitted from sensor devices reaches its intended destination accurately and without errors. By providing error checking, acknowledgment mechanisms, and flow control, TCP enhances the reliability of data transmission, which is particularly vital in sensor networks where accurate and timely information is imperative. However, while TCP offers reliability, it also introduces certain overhead due to its mechanisms, which might not be optimal for resource-constrained sensor devices. Thus, while TCP provides robustness, its implementation in sensor networks necessitates a careful balance between reliability and resource efficiency to optimize network performance.

REFERENCES

- https://www.geeksforgeeks.org/what-is-transmission-control-protocol-tcp/
- https://journals.sagepub.com/doi/10.1080/15501320601069861?icid=int.sj-full-text.similar-articles.5