

1. Introduction

a. Purpose

The purpose of this document is to provide a high level description of the TauNet System.

b. Scope

TauNet will be a secure communications network meant for a group of trusted and verified members.

c. Glossary

| Term | Definition |
|-------------|---|
| RC4 | Form of encryption |
| Node | A raspberry pi connected to the network. |
| User | An assumed human that is using a TauNet Node to communicate within the network. |
| System | The TauNet system/program |

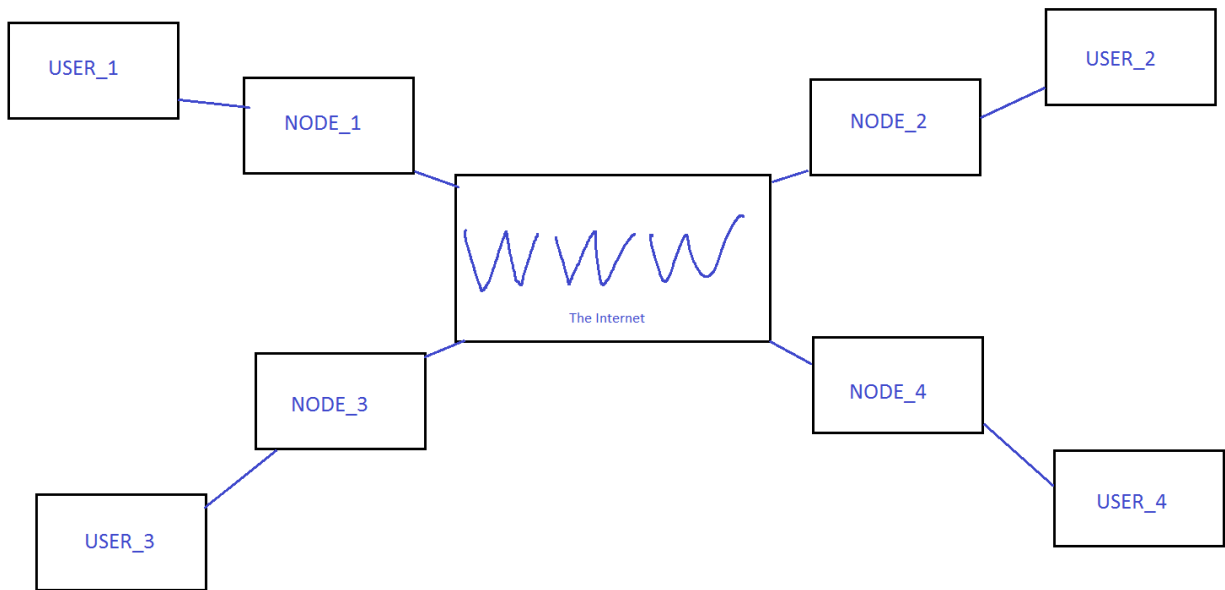
d. References

The TauNet communication protocol.

2. Overall Description

a. Environment

The diagram below is a rough illustration of how the components of the network will interact.



The TauNet system will be multiple nodes, connected over the internet with the ability to communicate with each other [other TauNet Nodes] via the TauNet communication protocol.

b. Functional Requirements

i. Requirements

1. Login
2. Logout
3. Send Message consisting of text.
4. View Contacts
5. Check Inbox
6. View Message

ii. Use Cases

1. *Login*

a. Actors:

User

b. Entry Conditions:

Raspberry pi is powered on and booted up.

User has started the TauNet application.

c. Exit Conditions:

User has been granted access to the functionality of TauNet and a list of options is displayed.

d. Flow of events:

- i. The system performs a login prompt by presenting a field for a username.
 - 1. The user enters their username via keyboard.
- ii. The system queries for the existence of the username.
- iii. If that username does not exist an "incorrect username" message is displayed and the system re-prompts for username.
- iv. In the case that the username IS associated with an active account, the system will prompt the user for a password.
 - 1. The user enters their password.
- v. The system compares the password with the persistent password for current username.
- vi. If the password is incorrect an "incorrect password" message is displayed and the system re-prompts for password
- vii. If the password matches the correct password the user is granted access.

2. *View Contacts*

- a. Actors:
 - User
- b. Entry Conditions:
 - User logged into their node and has been shown a list of options.
- c. Exit Conditions:
 - List of the user's contacts has been displayed.
- d. Flow of Events:
 - i. The User selects the corresponding number for view contacts from the list of available options.
 - 1. The system responds by Displaying the contacts list.

3. *Send Message*

- a. Actors:
 - User
- b. Entry Conditions:

User logged into their node. And has been shown a list of options.

c. Exit Conditions:

Confirmation of successful delivery or notification of failure.

d. Flow of events:

i. User selects the "Send Message" option.

1. The system responds by displaying a list of recipients (contact list) and prompting the user to choose a contact.

ii. The user selects a contact by inputting the corresponding number into the proceeding field.

1. The system the prompts the user to enter the desired message, up to 300 characters.

iii. The user types in their message and then hits enter/return.

1. If message sends, user is notified of successful send.
2. If message fails, user is notified of failure.

4. *Check Messages*

a. Actors:

User

b. Entry Conditions:

User logged into their account and list of available options is displayed.

c. Exit Conditions:

List of messages displayed.

d. Flow of Events:

i. The user selects the option for viewing messages by selecting the corresponding number.

1. The system displays all available messages.

5. *View Message*

a. Actors:

User

b. Entry Conditions:

Check messages use case must be invoked.

c. Exit Conditions:

The user is presented with the content of the message they have chosen to view.

d. Flow of Events:

- i. Presently on the screen, is a list of available messages with corresponding numbers. The user selects a message by inputting the corresponding number.

1. The system responds by displaying the full content of that message.

6. *Logout*

a. Actors:

User

b. Entry Conditions:

User must be logged in.

c. Exit Conditions:

User is logged out of account.

d. Flow of Events

- i. The user selects the corresponding number for the logout option.

1. The system terminates access to the network.

c. Non-Functional Requirements

i. Scalability

1. Minimum of 12 nodes.
2. 300 possible nodes?

ii. Reliability

1. Every node in the system should be online 90% of the time it is connected to the network.
2. Must be free from a single point of failure.

iii. Security

1. RC4 encrypted messages.
2. Access to network is by invite only.
3. Separate encryption key for each user.

iv. Environmental

1. Must be able to run on raspbian os.

v. Usability

1. Requires a keyboard, mouse and monitor connected to raspberry pi.

2. Assumes that user has a basic understanding of how to operate through a command line interface.