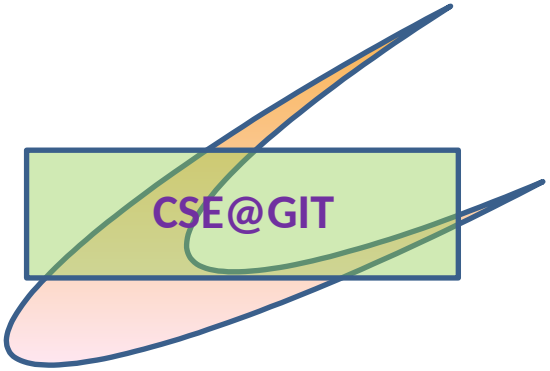
NCTUns Network Simulator



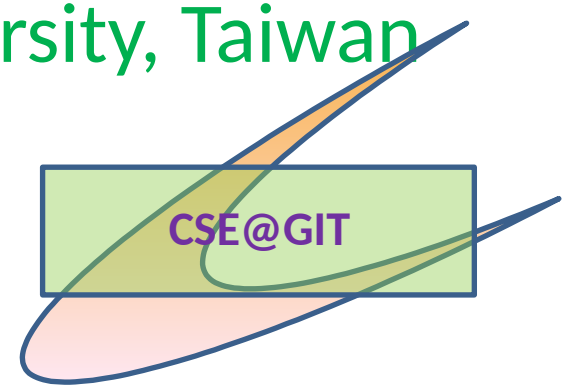
NCTUns Network Simulator

Shie-Yuan Wang, Ao-Jan Su, Kuo-Chiang Liao, Hsi-Yun Chen, and Meng-Chen Yu,

Network and System Laboratory,

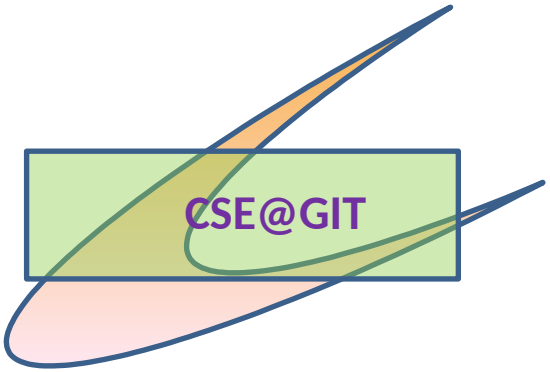
Department of Computer Science and Information Engineering,

National Chiao Tung University, Taiwan



Content:

1. A high-level view of NCTUns network simulator
2.
3.
4.



A high-level view of NCTUns network simulator

A high-level view of NCTUns network simulator

- **Support for Various Networks**
- **Support for Various Networking Devices**
- **Support for Various Network Protocols**

A high-level view of NCTUns network simulator

- **Support for Various Networks**

  Wired networks with fixed nodes and point-to-point links

  Wireless networks with mobile nodes and IEEE 802.11 (b)

  Wireless network interfaces

- **Support for Various Networking Devices**
- **Support for Various Network Protocols**

A high-level view of NCTUns network simulator

- **Support for Various Networks**
- **Support for Various Networking Devices**

  Ethernet hubs, switches, routers, hosts,

  IEEE 802.11 wireless access points and interfaces

- **Support for Various Network Protocols**

A high-level view of NCTUns network simulator

- **Support for Various Networks**
- **Support for Various Networking Devices**
- **Support for Various Network Protocols**

  IEEE 802.3 CSMA/CD MAC, IEEE 802.11 (b), CSMA/CA MAC

  Learning bridge, spanning tree

  IP, RIP, OSPF, UDP, TCP, HTTP, FTP, Telnet

Major modules of NCTUns network simulator

Major modules of NCTUns network simulator

- **GUI**
- **Simulation job dispatcher**
- **Coordinator**

Major modules of NCTUns network simulator

- **GUI**

  Draw network topologies

  Configuring the protocol modules used inside a node

  Specifying the moving path of mobile nodes

  Plot network performance graphs

  Play back the animation of a logged packet transfer trace

- **Simulation job dispatcher**
- **Coordinator**

Major modules of NCTUns network simulator

- **GUI**
- **Simulation job dispatcher**

  Manage and use multiple simulation servers at the same time to increase aggregate simulation throughput (Server)

- **Coordinator**

Major modules of NCTUns network simulator

- **GUI**
- **Simulation job dispatcher**
- **Coordinator**

  Coordinator process is alive as long as the simulation machine is alive

  Registers itself with the dispatcher to join the dispatcher's simulation machine

  Forks a simulation server process to simulate the specified network and protocols to execute a job

  Communicates with the dispatcher and the GUI program on behalf of the simulation server process

  Exchanges message between the simulation server process and the GUI program

Algorithm for the experiment

- Declare the required header files limits.h, unistd.h, iostream.h
- Define POSIX standard constants
- Menu driven code for compile or run time limits checking
- Use macros for querying compile time limits
- Use sysconf for process related limits checking and pathconf or fpathconf for File related limits checking

Pseudo Code / Outline of the Algorithm

For Compile Time Limits:

If the macro is defined output it's value

else a message indicating that it is not defined

Ex:

```
#ifdef _POSIX_OPEN_MAX
    cout<<"Max no of files simultaneously opened"<< POSIX_OPEN_MAX
 else
    cout<<"Macro _POSIX_OPEN_MAX not defined"
```

Pseudo Code / Outline of the Algorithm

For Run Time Limits:

Use sysconf for process related limits and

pathconf or fpathconf for file related limits

Pseudo Code / Outline of the Algorithm

Working with sysconf function:

```
if ((res=sysconf(_SC_CLK_TCK))==-1)
      perror("sysconf");
else
      cout << "Number of Clock Ticks is :  "
            << res << endl;
```

Sample Run

Compile Time Limits:

Max number of child processes	24
Max path length	80
Max number of chars in file name	255
Max number of open files per process	20

RunTime Limits:

Number of clock ticks	100
Max path length	255
Max number of chars in file name	255
Max number of open files per process	20

Learning Outcomes

At the end of the session, students should be able to :

1) Have a high-level view about the NCTUns network simulator

2) Install the NCTUns 1.0 network simulator package

3) Run a simulation

4) Compare compile time and run time limits. [L 4]