This is the best assignment solution uploaded with some modification of the following student:

Name**: Samreen Laghari**
Student Id**: ms110400048**

Question No. 1:

a) **Consider a point-to-point link 2 km in length. At what bandwidth would propagation delay (at a speed of 2×10^8 m/s) equal transmit delay for 100-byte packets? What about 512-byte packets?**

b) **How long does it take to transmit x KB over a y-Mbps link? Give your answer as a ratio of x and y.**

c) **Suppose that a certain communications protocol involves a per-packet overhead of 100 bytes for headers and framing. We send 1 million bytes of data using this protocol; however, one data byte is corrupted and the entire packet containing it is thus lost. Give the total number of overhead + loss bytes for packet data sizes of 1000, 5000, 10,000, and 20,000 bytes. Which size is optimal?**

Solution Part (a):

<u>100-Byte Packets</u>

Given Data:
Distance = 2 Km = 2×10^3 m
SpeedOfLight = 2×10^8 m/s
Packet Size = 100 bytes = 100×8 = 800 bits
Propagation Delay = Transmit Delay
To Find:
Bandwidth = X =?
Solution:
According to the Formula given in "Computer Networks: A Systems Approach" by "Larry L. Peterson and Bruce S. Davie";

$$Transmit\ Delay = \frac{Size}{Bandwidth}$$ and

$$\text{Propogation Delay} = \frac{\text{Distance}}{\text{SpeedOfLight}}$$

Putting the given values in the above equation; we get

$$Transmit\ Delay = \frac{800\ b}{X\ b/s}$$

$$\text{Propogation Delay} = \frac{2\times10^3 m}{2\times10^8 m/s} = 1\times10^{-5} s = 10\times10^{-6} s = 10\mu s$$

$\text{Propogation Delay} = 10\mu s$

Equating the Transmit Delay and the Propagation Delay; we get
Propagation Delay = Transmit Delay

$$\frac{800\ b}{X\ b/s} = 10\times10^{-6}$$

$X\times10\times10^{-6} = 800$

$$X = \frac{800}{10\times10^{-6}} = 80\times10^6\ bit/s = 80\ Mbit/s$$

$X = 80\ Mbit/s$

Hence, the propagation delay equals transmit delay for 100-byte packets at the bandwidth of 80Mbit/s.

512-Byte Packets

Given Data:
Distance = 2 Km = 2×10^3 m
SpeedOfLight = 2×10^8 m/s
Packet Size = 512 bytes = 512 × 8 = 4096 bits
Propagation Delay = Transmit Delay
To Find:
Bandwidth = X =?
Solution:
According to the Formula given in "Computer Networks: A Systems Approach" by "Larry L. Peterson and Bruce S. Davie";

$$Transmit\ Delay = \frac{Size}{Bandwidth} \qquad \text{and}$$

$$\text{Propogation Delay} = \frac{\text{Distance}}{\text{SpeedOfLight}}$$

Putting the given values in the above equation; we get

$$Transmit\ Delay = \frac{4096\ b}{X\ b/s}$$

$$\text{Propogation Delay} = \frac{2\times10^3 m}{2\times10^8 m/s} = 1\times10^{-5} s = 10\times10^{-6} s = 10\mu s$$

$\text{Propogation Delay} = 10\mu s$

Equating the Transmit Delay and the Propagation Delay; we get
Propagation Delay = Transmit Delay

$$\frac{4096\ b}{X\ b/s}=10\times10^{-6}$$

$$X\times10\times10^{-6}=4096$$

$$X=\frac{4096}{10\times10^{-6}}=409.6\times10^{6}\ bit/s=409.6\ Mbit/s$$

$$X=409.6\ Mbit/s$$

Hence, the propagation delay equals transmit delay for 512-byte packets at the bandwidth of 409.6 Mb/s.

Solution Part (b):

Number of bytes in a KB = 1024 bytes
Number of bits in a byte = 8 bits

$$x\ KB=1024\times8\times x\ bits$$

Number of bits in a Mb = 10^6 bits

$$y\ Mbps=10^{6}\ y\ bps$$

The time "t" taken to transmit x KB over a y-Mbps link can be given as follows:

$$t=\frac{8\times1024\times x}{y\times10^{6}}$$

$$t=\frac{8192\times x}{y\times10^{6}}$$

$$t=\frac{8192\times10^{-6}\times x}{y}$$

$$t=\frac{8.192\times10^{-3}\times x}{y}$$

$$t=\frac{8.192x}{y}ms$$

Hence, the taken to transmit x KB over a y-Mbps link is found to be 8.912x/y ms.

Solution Part (c):

Given Data:
Data Size of Packet = D
Per-Packet Overhead = 100 bytes
Number of bytes sent using the protocol = 10^6

Number of Packets needed = N = $\frac{10^6}{D}$

Overhead = $100\times N$

Loss = D

$$Overhead + Loss = 100 \times N + D$$

$$Overhead + Loss = 100 \times \left(\frac{10^6}{D} \right) + D$$

To Find:

Total number of overhead + loss bytes for packet data size, D=1000 =?
Total number of overhead + loss bytes for packet data size, D=5000 =?
Total number of overhead + loss bytes for packet data size, D=10,000 =?
Total number of overhead + loss bytes for packet data size, D=20,000 =?

Solution:

Overhead + Loss bytes for D=1000

According to the formula given above:

$$Overhead + Loss = 100 \times \left(\frac{10^6}{D} \right) + D$$

Putting the given values, we get;

$$Overhead + Loss = 100 \times \left(\frac{10^6}{1000} \right) + 1000$$

$$Overhead + Loss = 100 \times 10^3 + 1000$$

$$Overhead + Loss = 100 \times 1000 + 1000 = 101000 \quad \Rightarrow \quad Overhead + Loss = 101000$$

$Overhead + Loss$ bytes for Packet Data Size of $1000 = 101000$

Overhead + Loss bytes for D=5000

According to the formula given above:

$$Overhead + Loss = 100 \times \left(\frac{10^6}{D} \right) + D$$

Putting the given values, we get;

$$Overhead + Loss = 100 \times \left(\frac{10^6}{5000} \right) + 5000$$

$$Overhead + Loss = 100 \times \left(\frac{1000}{5} \right) + 5000$$

$$Overhead + Loss = 100 \times 200 + 5000 = 25000 \quad \Rightarrow \quad Overhead + Loss = 25000$$

$Overhead + Loss$ bytes for Packet Data Size of $5000 = 25000$

Overhead + Loss bytes for D=10,000

According to the formula given above:

$$Overhead + Loss = 100 \times \left(\frac{10^6}{D}\right) + D$$

Putting the given values, we get;

$$Overhead + Loss = 100 \times \left(\frac{10^6}{10000}\right) + 10000$$

$$Overhead + Loss = 10000 + 10000 = 20000 \quad \Rightarrow \quad Overhead + Loss = 20,000$$

$Overhead + Loss$ bytes for Packet Data Size of $10,000 = 20,000$

Overhead + Loss bytes for D=20,000

According to the formula given above:

$$Overhead + Loss = 100 \times \left(\frac{10^6}{D}\right) + D$$

Putting the given values, we get;

$$Overhead + Loss = 100 \times \left(\frac{10^6}{20000}\right) + 20000$$

$$Overhead + Loss = 100 \times \left(\frac{100}{2}\right) + 20000$$

$$Overhead + Loss = 100 \times 50 + 20000$$

$$Overhead + Loss = 5000 + 20000 = 25000 \quad \Rightarrow \quad Overhead + Loss = 25,000$$

$Overhead + Loss$ bytes for Packet Data Size of $20,000 = 25,000$

Summarizing the results in a table, we have;

Packet data Size (D)	Overhead + Loss Bytes
1000	101,000
5000	25,000
10,000	20,000 → Optimal Packet data Size
20,000	25,000

Hence, the total number of overhead + loss bytes for packet data sizes of 1000, 5000, 10,000, and 20,000 bytes are found to be 101,000 bytes, 25,000 bytes, 20,000 bytes, 25,000 bytes respectively. Since, the packet data size of 10,000 has the lowest number of Overhead + Loss Bytes i.e. 20,000 bytes, it is the optimal packet data size.

Question No. 2:

a) Calculate the latency (from first bit sent to last bit received) for the following:
   a. 10-Mbps Ethernet with a single store-and-forward switch in the path, and a packet size of 5000 bits. Assume that each link introduces propagation delay of 10 µs and that the switch begins retransmitting immediately after it has finished receiving the packet.
   b. Same as (a) but with three switches.

b) The 10 Mbps Ethernet segment can be as long as 2500 meters. If the 10 Mbps Ethernet frame specifications stay the same, how does in increase in supported bandwidth affect the segment length?

Solution Part (a):

(a) Latency With One Switch

Given Data:
Bandwidth = 10-Mbps
Packet Size = 5000 bits
Propagation Delay = 10 µs
Number of switches = 1
Number of links = 2

To Find:
Latency =?

Solution:

The transmission Delay can be calculated as follows:

$$Transmit\ Delay = \frac{Packet\ Size}{Bandwidth}$$

Putting the given values, we get;

$$Transmit\ Delay = \frac{5000}{10\times10^{6}} = \frac{5}{10\times10^{3}} = \frac{1}{2\times10^{3}} = 0.0005 = 500\times10^{-6} = 500\mu s$$

The Latency can be calculated from the formula as follows:

$$\text{Latency} = \text{Transmit Delay for each link} + \text{Propagation Delay for each link}$$

Putting the given values of Propagation and Transmit delay, we get;

$$Latency = 500\mu s \text{ for each link} + 10\mu s \text{ for each link}$$

$$Latency = 500\mu s\times2 + 10\mu s\times2 = 1000\mu s + 20\mu s = 1020\mu s$$

$$Latency = 1020\mu s$$

Hence, latency (from first bit sent to last bit received) is found to be 1020 µs

(b) Latency With Three Switches

Given Data:
Bandwidth = 10-Mbps
Packet Size = 5000 bits
Propagation Delay = 10 µs
Number of switches = 3
Number of links = 4

To Find:
Latency =?

Solution:

The transmission Delay can be calculated as follows:

$$Transmit\ Delay = \frac{Packet\ Size}{Bandwidth}$$

Putting the given values, we get;

$$Transmit\ Delay = \frac{5000}{10\times10^{6}} = \frac{5}{10\times10^{3}} = \frac{1}{2\times10^{3}} = 0.0005 = 500\times10^{-6} = 500\mu s$$

The Latency can be calculated from the formula as follows:

$$\text{Latency} = \text{Transmit Delay for each link} + \text{Propagation Delay for each link}$$

Putting the given values of Propagation and Transmit delay, we get;

$$Latency = 500\mu s \text{ for each link} + 10\mu s \text{ for each link}$$

$$Latency = 500\mu s\times4 + 10\mu s\times4 = 2000\mu s + 40\mu s = 2040\mu s = 2.04\times10^{-3} = 2.04ms$$

$$Latency = 2.040ms$$

Hence, latency (from first bit sent to last bit received) is found to be 2.04 ms.

Solution Part (b):

Two factors affect the segment length while increasing the supported bandwidth, i.e. the “minimum frame length” and the “contention period”. Increasing the supported bandwidth will result in less transmission time of a 64-byte frame however; the “contention period” should remain the same. By keeping the “contention period” the same, and if the propagation time of a bit stays the same, the length of the segment drops down. Hence, increase in supported bandwidth will decrease the segment length.

Question No. 3:

Read Beej's guide to Network Programming and modify the talker/listener (talker.c and listener.c) according to the following specifications (Listener and talker are communicating using UDP socket):

Listener:

1. The listener should run in continuous mode, i.e., it should handle one talker after another (Make listener iterative server)
2. The listener will bind on port # 4500
3. The listener should maintain a log named "TalkerMsgLog.txt" of all the talkers and their messages. The log should contain the following information
    a. IP address of the talker
    b. Port number of the talker
    c. Message sent by the talker
4. When a listener receives a message from a talker, it save the talkers message in the log and also sends a reply to the talker containing the following well message:
    "Welcome: TALKER-IP-ADDRESS, TALKER-PORTNO"

5. When a listener receives a message from a talker, it saves the talker's message in the log "FirstDataMsg: TALKER-IP-ADDRESS, TALKER-PORTNO: this is first formal data message from Talker" and also sends a reply to the talker containing the following message:
    "IDMsg: Your First Data Message is received, please send your name and ID No"
6. When the listener receives a message from the talker, it saves the talker's message in the log "IDMsgReply: TALKER-IP-ADDRESS, TALKER-PORTNO: I am Muhammad – MS090400001" and also sends a reply to the talker containing the following message:
    "TermMsg: Your Name and ID is logged on Listener, communication terminated"
7. Listener then waits for some new talker and does not terminate.

Talker:

1. A talker takes the following command line argument from the user:
    a. IP address of the listener
    b. Port No. of Listener
2. The command line arguments entered by the user will be : ./talker 127.0.0.1 4500
3. Create a log file "ListernerMsgLog.txt"
4. Talker formats a message and sends to the listener on a IP address and port taken as a command line argument.
5. When the talker receives the welcome message as a reply from the listener, it displays the message on the screen and writes it in the log file.
6. Talker formats a message and sends to the listener: I am "Your Name – MS**04000**"(ID).
7. When the talker receives the TermMsg message from the listener, it displays the message on the screen and writes in the log file, terminates the communication and exit.

Solution:

Listener.C

```
/*
** listener.c -- a datagram sockets "server" demo
*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define MYPORT 4500 // the port users will be connecting to
#define MAXBUFLEN 100
int main(void)
{
int sockfd;
struct sockaddr_in my_addr; // my address information
struct sockaddr_in their_addr; // connector's address information
int addr_len, numbytes;
char buf[MAXBUFLEN];
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
perror("socket");
exit(1);
}
my_addr.sin_family = AF_INET; // host byte order
my_addr.sin_port = htons(MYPORT); // short, network byte order
my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my IP
memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct
if (bind(sockfd, (struct sockaddr *)&my_addr,
sizeof(struct sockaddr)) == -1) {
perror("bind");
exit(1);
}
addr_len = sizeof(struct sockaddr);
if ((numbytes=recvfrom(sockfd, buf, MAXBUFLEN-1 , 0,
(struct sockaddr *)&their_addr, &addr_len)) == -1) {
perror("recvfrom");
exit(1);
```

```
}
printf("got packet from %s\n",inet_ntoa(their_addr.sin_addr));
printf("packet is %d bytes long\n",numbytes);
buf[numbytes] = '\0';
printf("packet contains \"%s\"\n",buf);
close(sockfd);
return 0;
}
```

Talker.C

```
/*
** talker.c -- a datagram "client" demo
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>

#define SERVERPORT "4950"        // the port users will be connecting to

int main(int argc, char *argv[])
{
        int sockfd;
        struct addrinfo hints, *servinfo, *p;
        int rv;
        int numbytes;

        if (argc != 3) {
                fprintf(stderr,"usage: talker hostname message\n");
                exit(1);
        }

        memset(&hints, 0, sizeof hints);
        hints.ai_family = AF_UNSPEC;
        hints.ai_socktype = SOCK_DGRAM;

        if ((rv = getaddrinfo(argv[1], SERVERPORT, &hints, &servinfo)) != 0) {
```

```
                fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(rv));
                return 1;
        }

        // loop through all the results and make a socket
        for(p = servinfo; p != NULL; p = p->ai_next) {
                if ((sockfd = socket(p->ai_family, p->ai_socktype,
                                p->ai_protocol)) == -1) {
                        perror("talker: socket");
                        continue;
                }

                break;
        }

        if (p == NULL) {
                fprintf(stderr, "talker: failed to bind socket\n");
                return 2;
        }

        if ((numbytes = sendto(sockfd, argv[2], strlen(argv[2]), 0,
                        p->ai_addr, p->ai_addrlen)) == -1) {
                perror("talker: sendto");
                exit(1);
        }

        freeaddrinfo(servinfo);

        printf("talker: sent %d bytes to %s\n", numbytes, argv[1]);
        close(sockfd);

        return 0;
}
```

Question No. 4:

Read the attached research paper entitled "Design and Development of a UDP- Based Connection-Oriented Multi-Stream One-to-Many Communication Protocol" with this assignment very carefully. Summarize the research paper into your own words (450-500) in the following headings:

1. Proposed technique
2. Advantages / Benefits
3. Disadvantages / Drawbacks

4. Suggestions & Recommendations

Solution:

Please! Consult the mentioned Research Paper.

Question No. 5:

Read the attached research paper entitled "A Survey of Packet Loss Recovery Techniques for Streaming Audio" with this assignment very carefully, and answer the following questions:

1. Classify the recovery techniques discussed in this paper and write a brief note.
2. Give the comparison of the Forward Error Correction Techniques to repair losses of data during transmission, discussed in this paper.
3. Give the comparison of the Error Concealment Techniques, discussed in this paper.

Solution:

Please! Consult the mentioned Research Paper.

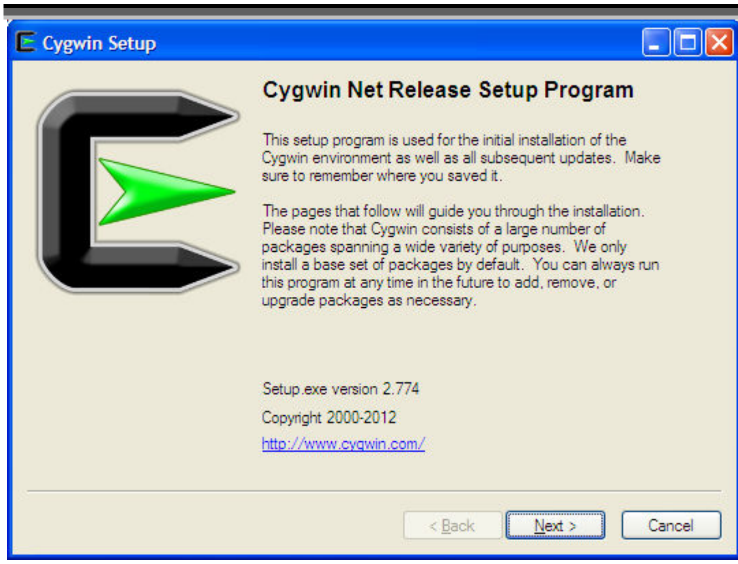
Question No. 6:

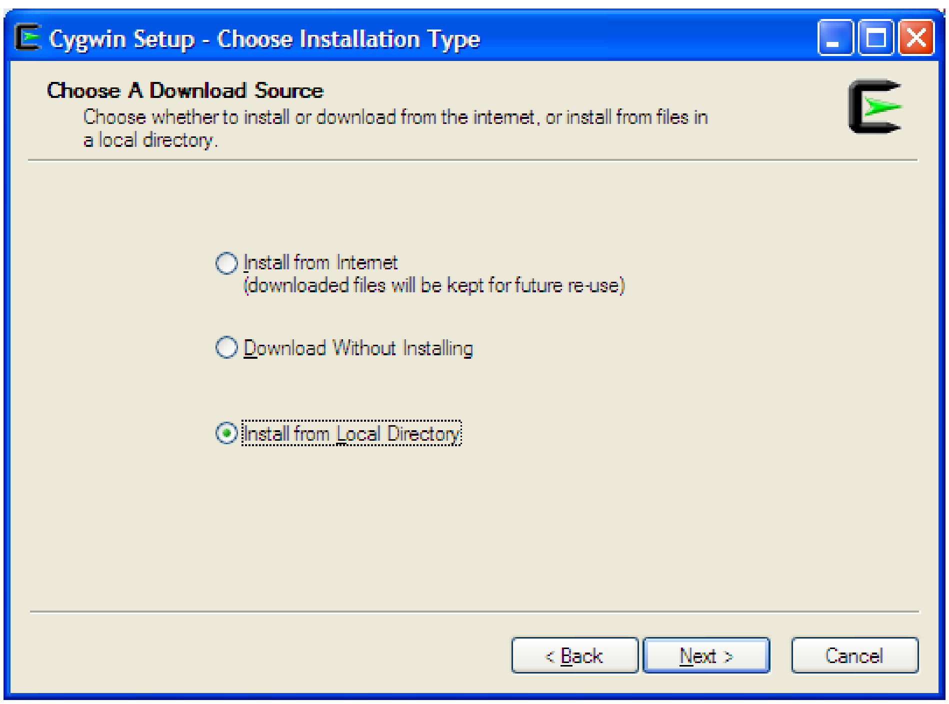
Explore Network Simulator (NS 2).

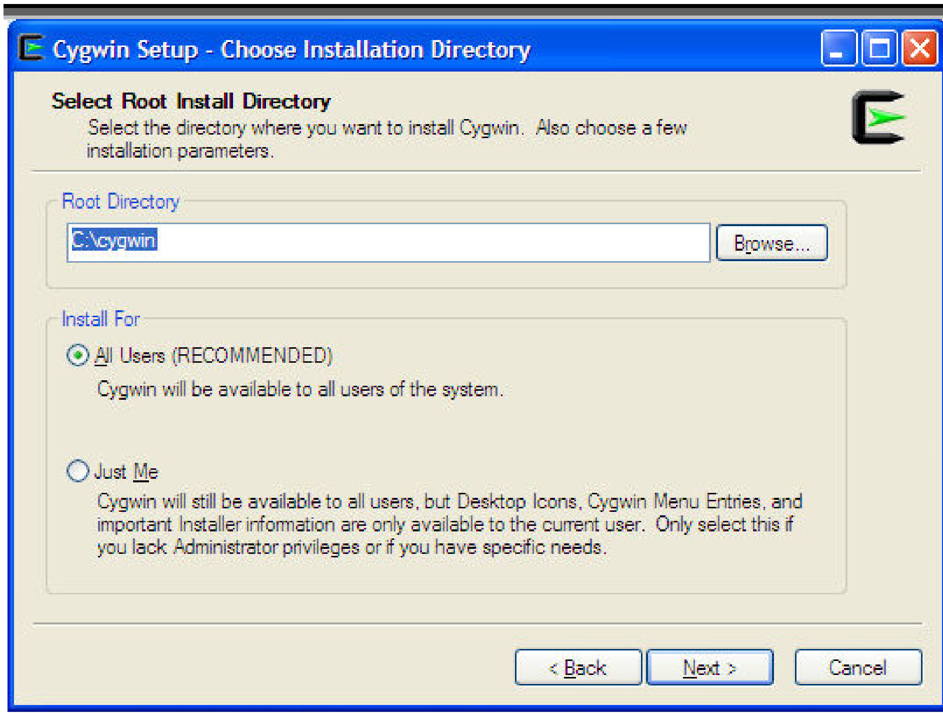
1. Installation & Configurations
2. Show screen shots of installation and configuration on your system which must show your identity with your system.
3. Write a brief summary about installations and configurations on your system.
4. Use simulation files exist in NS 2 library and show also its NAM interface.

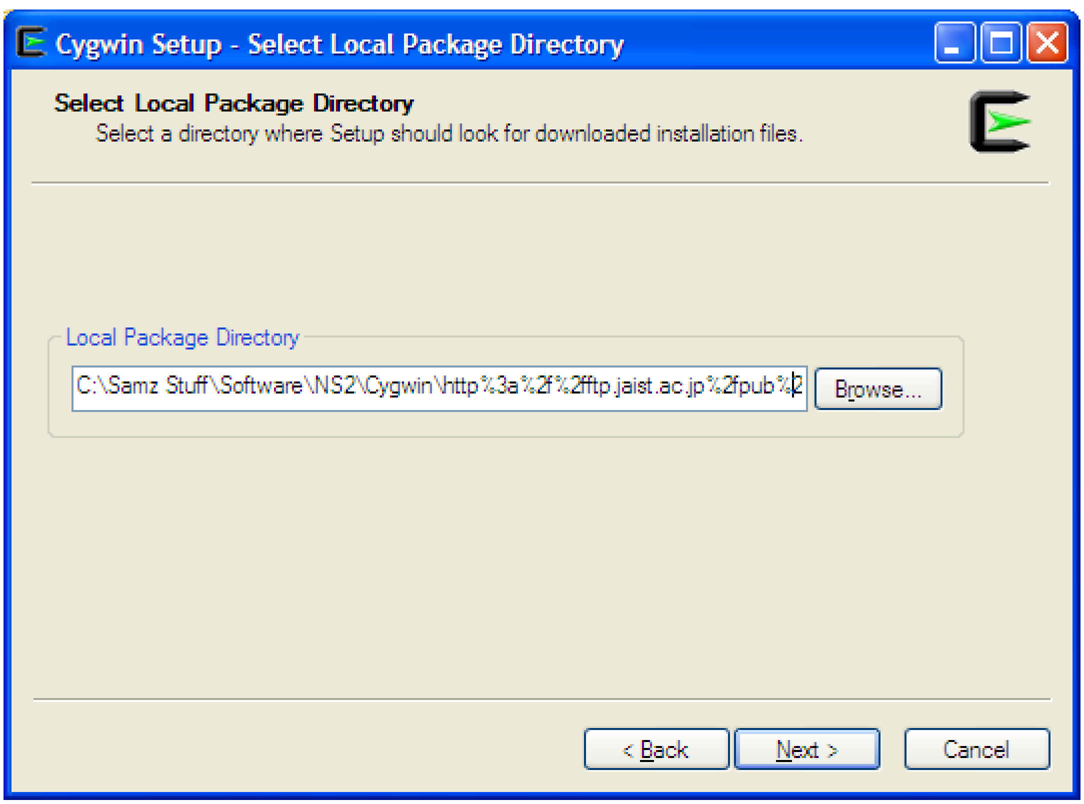
Solution:

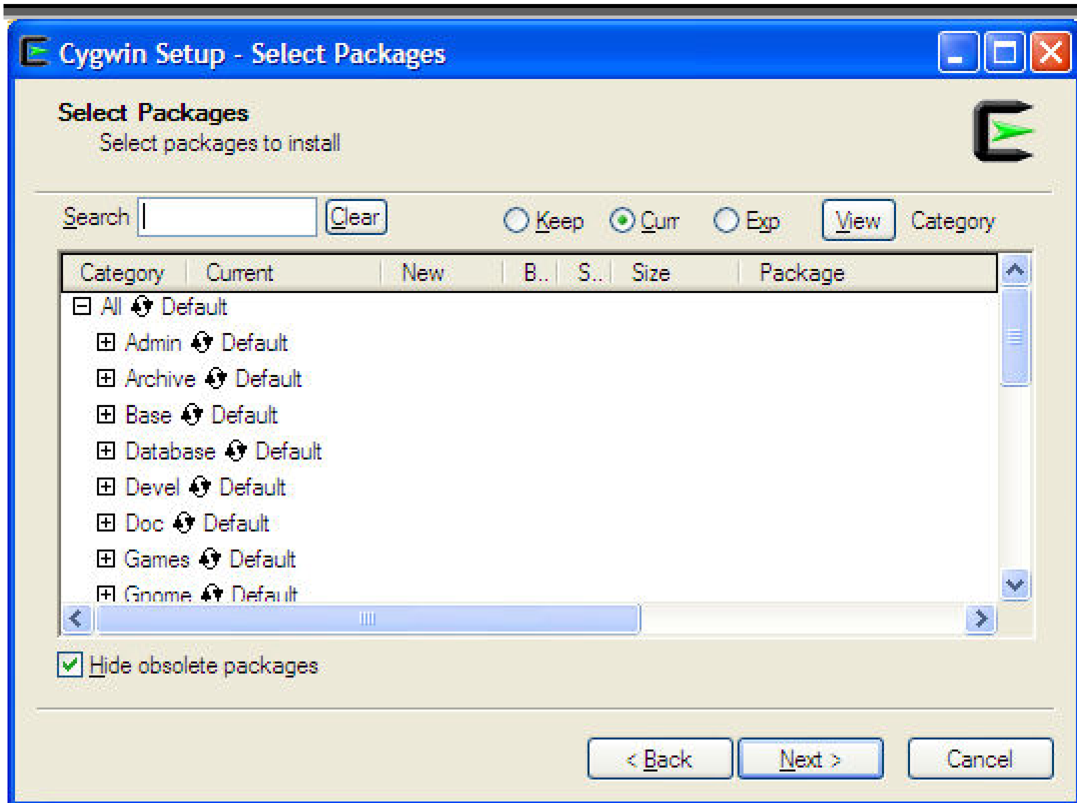
Screenshots of Installation and Configuration

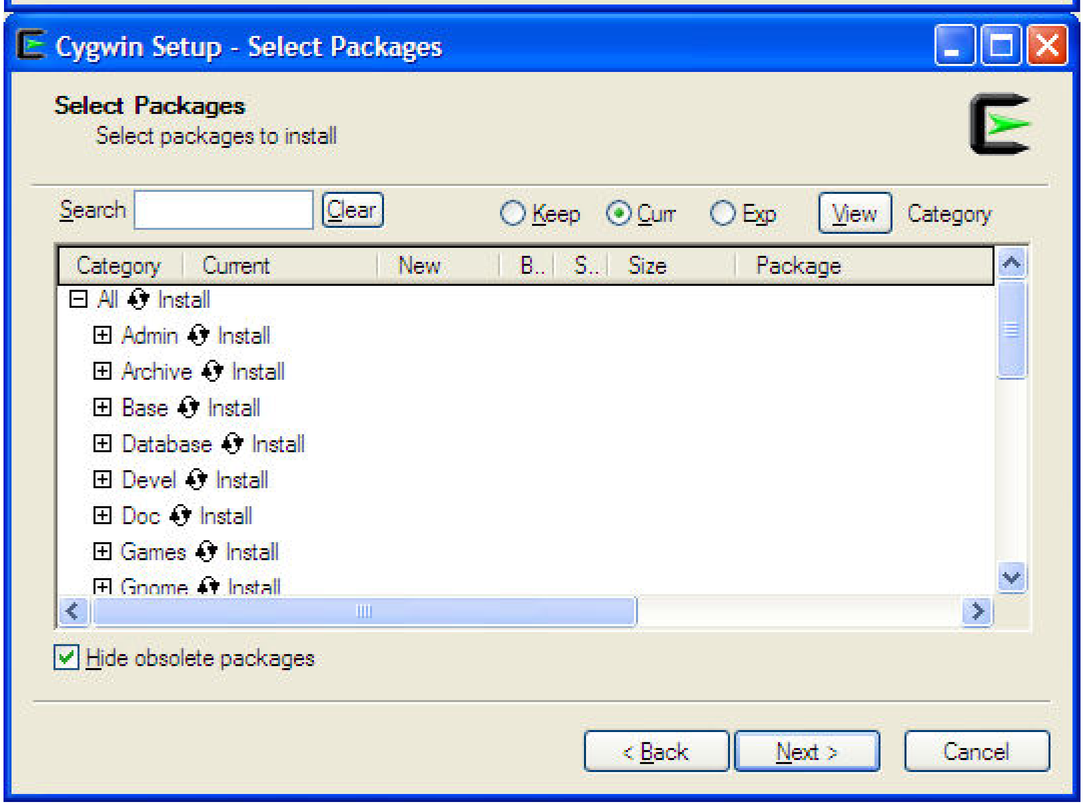

Cygwin Setup
Cygwin Net Release Setup Program
This setup program is used for the initial installation of the Cygwin environment as well as all subsequent updates. Make sure to remember where you saved it.
The pages that follow will guide you through the installation. Please note that Cygwin consists of a large number of packages spanning a wide variety of purposes. We only install a base set of packages by default. You can always run this program at any time in the future to add, remove, or upgrade packages as necessary.
Setup.exe version 2.774
Copyright 2000-2012
http://www.cygwin.com/
< Back
Next >
Cancel

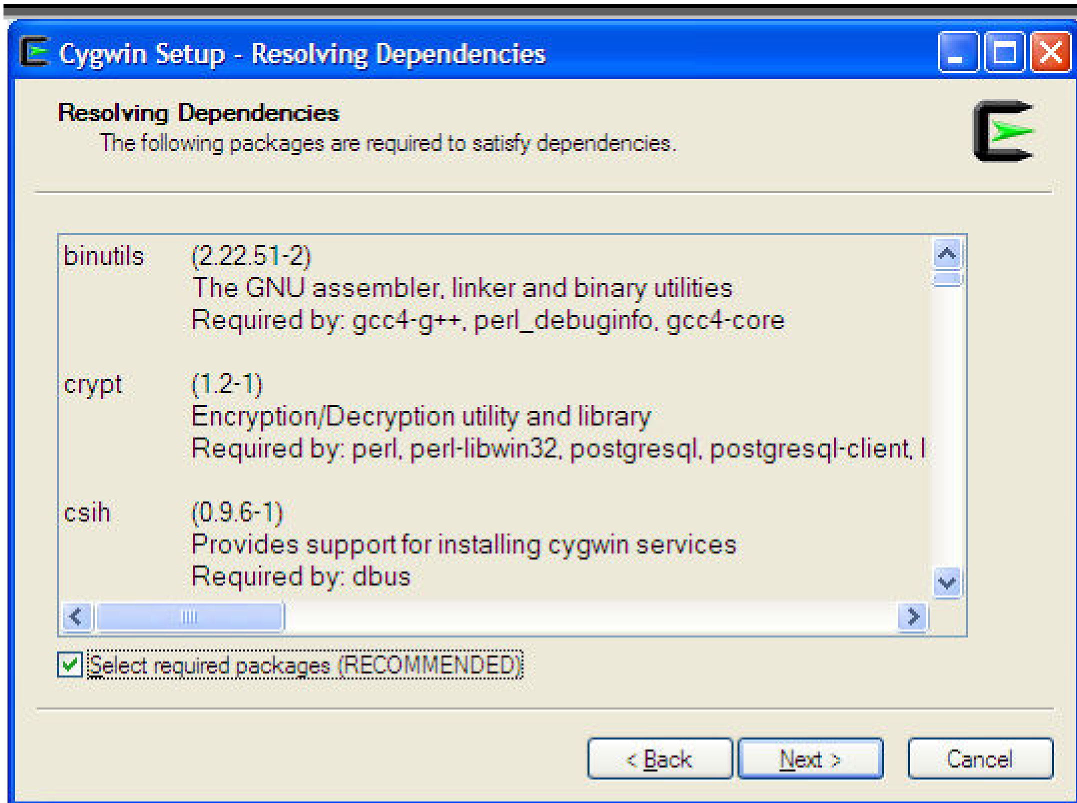

Cygwin Setup - Choose Installation Type
Choose A Download Source
Choose whether to install or download from the internet, or install from files in a local directory.
Install from Internet
(downloaded files will be kept for future re-use)
Download Without Installing
Install from Local Directory
< Back
Next >
Cancel

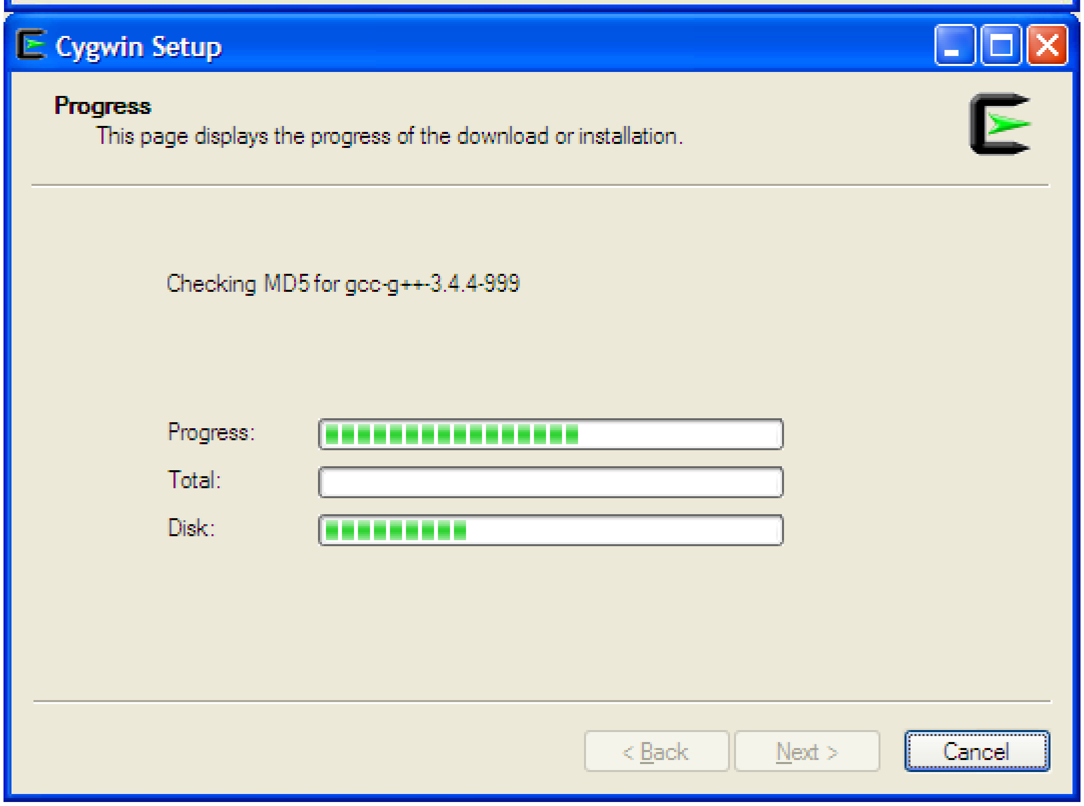

Cygwin Setup - Choose Installation Directory
Select Root Install Directory
Select the directory where you want to install Cygwin. Also choose a few installation parameters.
Root Directory
C:\cygwin
Browse...
Install For
All Users (RECOMMENDED)
Cygwin will be available to all users of the system.
Just Me
Cygwin will still be available to all users, but Desktop Icons, Cygwin Menu Entries, and important Installer information are only available to the current user. Only select this if you lack Administrator privileges or if you have specific needs.
< Back
Next >
Cancel

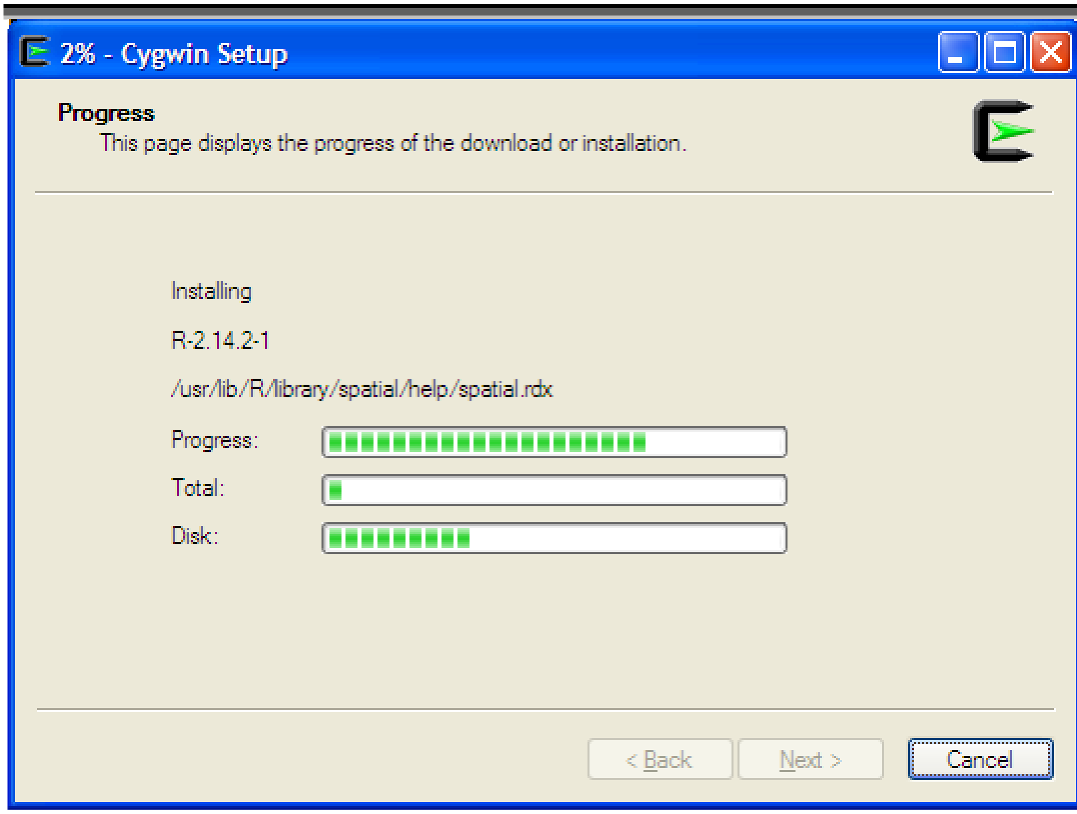

Cygwin Setup - Select Local Package Directory
Select Local Package Directory
Select a directory where Setup should look for downloaded installation files.
Local Package Directory
C:\Samz Stuff\Software\NS2\Cygwin\http%3a%2f%2fftp.jaist.ac.jp%2fpub%2
Browse...
< Back
Next >
Cancel

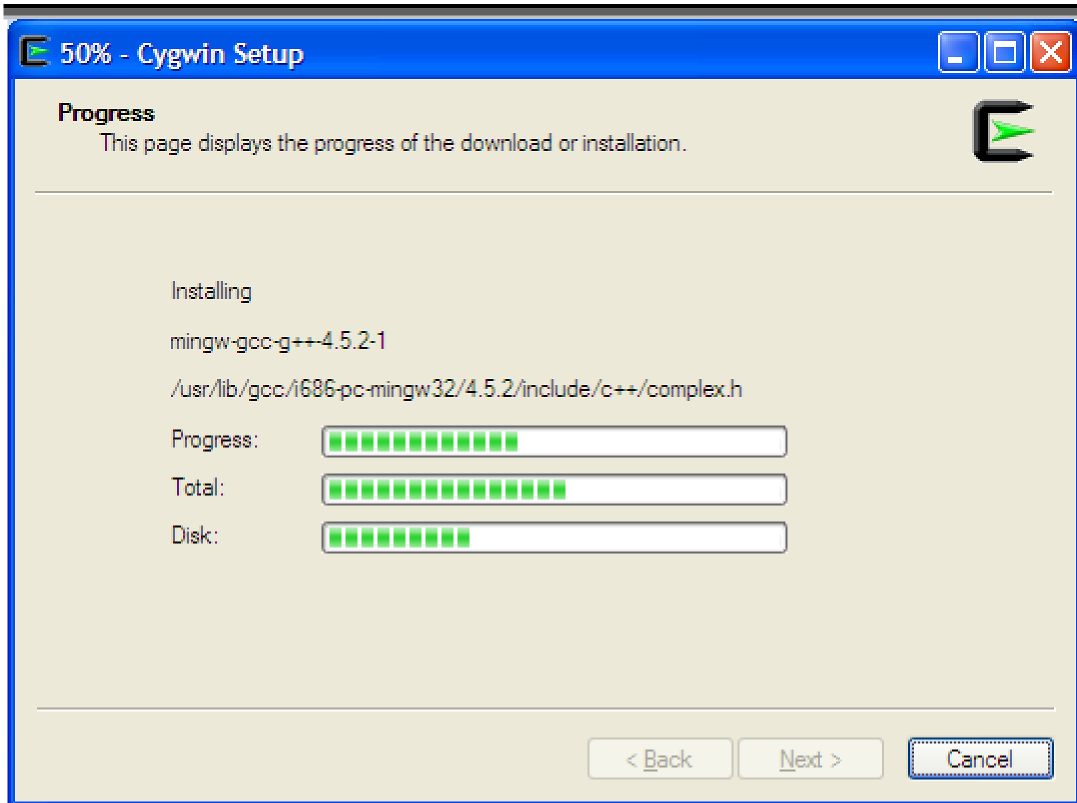

Cygwin Setup - Select Packages
Select Packages
Select packages to install
Search
Clear
Keep
Curr
Exp
View
Category
Category
Current
New
B..
S..
Size
Package
All Default
Admin Default
Archive Default
Base Default
Database Default
Devel Default
Doc Default
Games Default
Hide obsolete packages
< Back
Next >
Cancel

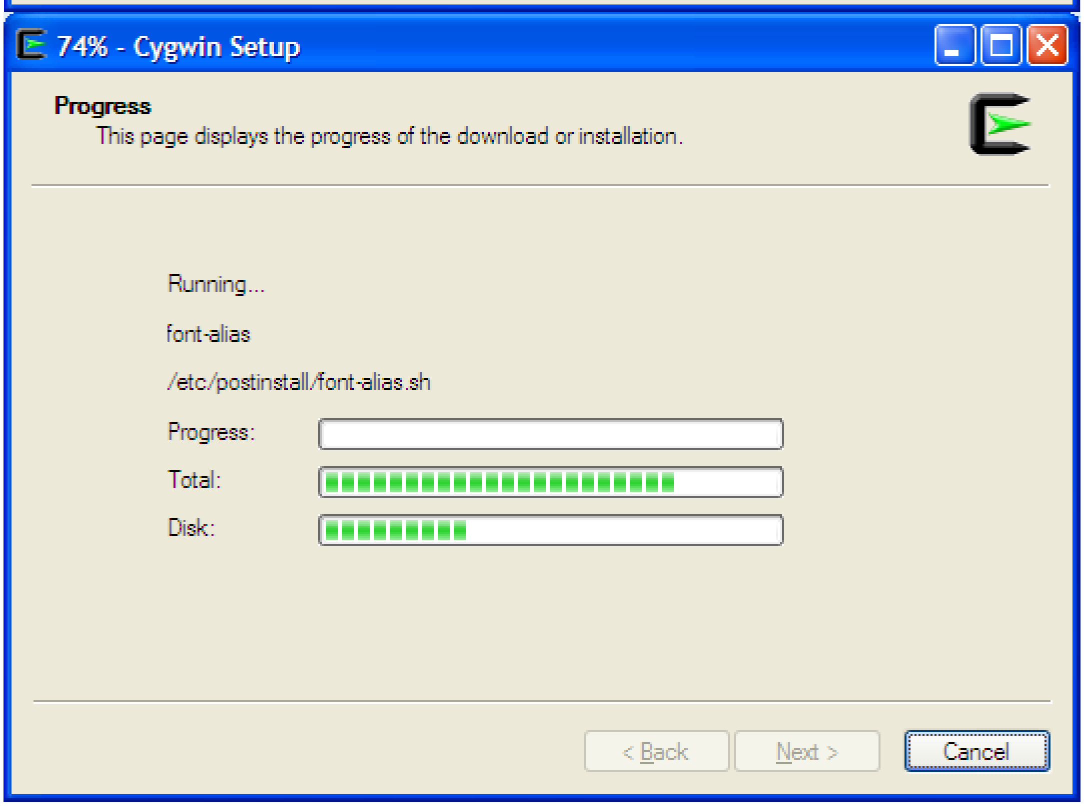

Cygwin Setup - Select Packages
Select Packages
Select packages to install
Search
Clear
Keep
Curr
Exp
View
Category
Category
Current
New
B..
S..
Size
Package
All Install
Admin Install
Archive Install
Base Install
Database Install
Devel Install
Doc Install
Games Install
Hide obsolete packages
< Back
Next >
Cancel

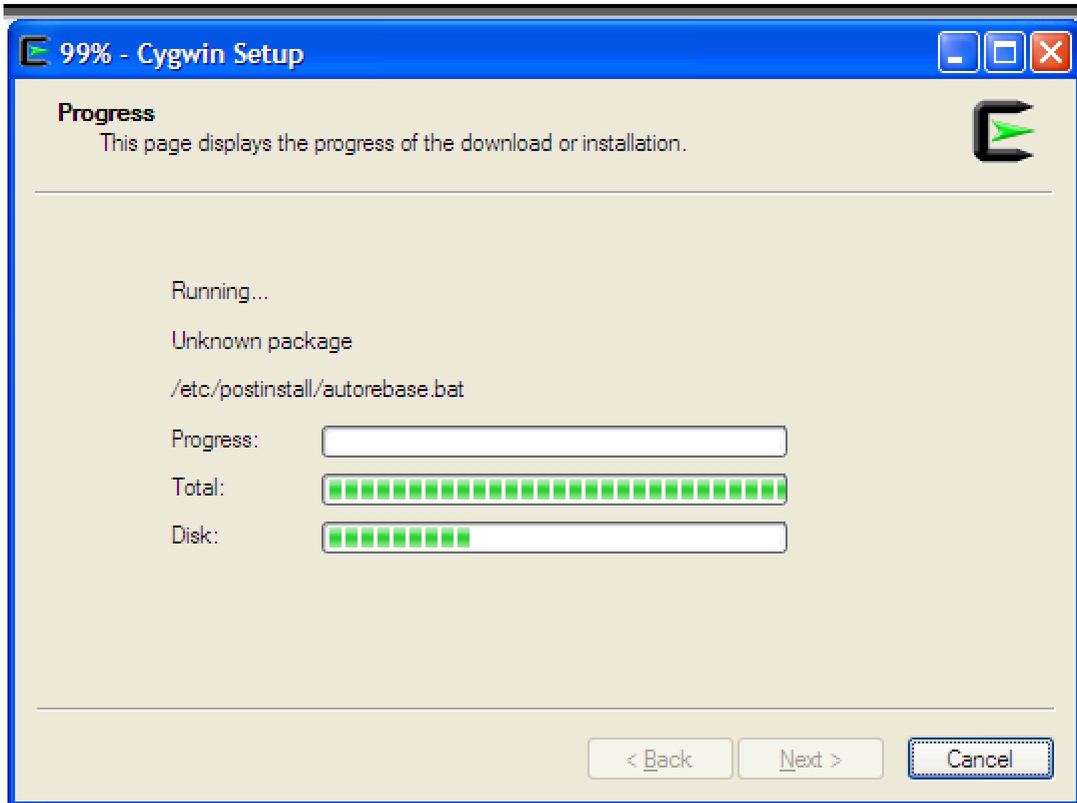

Cygwin Setup - Resolving Dependencies
Resolving Dependencies
The following packages are required to satisfy dependencies.
binutils (2.22.51-2)
The GNU assembler, linker and binary utilities
Required by: gcc4-g++, perl_debuginfo, gcc4-core
crypt (1.2-1)
Encryption/Decryption utility and library
Required by: perl, perl-libwin32, postgresql, postgresql-client, l
csih (0.9.6-1)
Provides support for installing cygwin services
Required by: dbus
Select required packages (RECOMMENDED)
< Back
Next >
Cancel

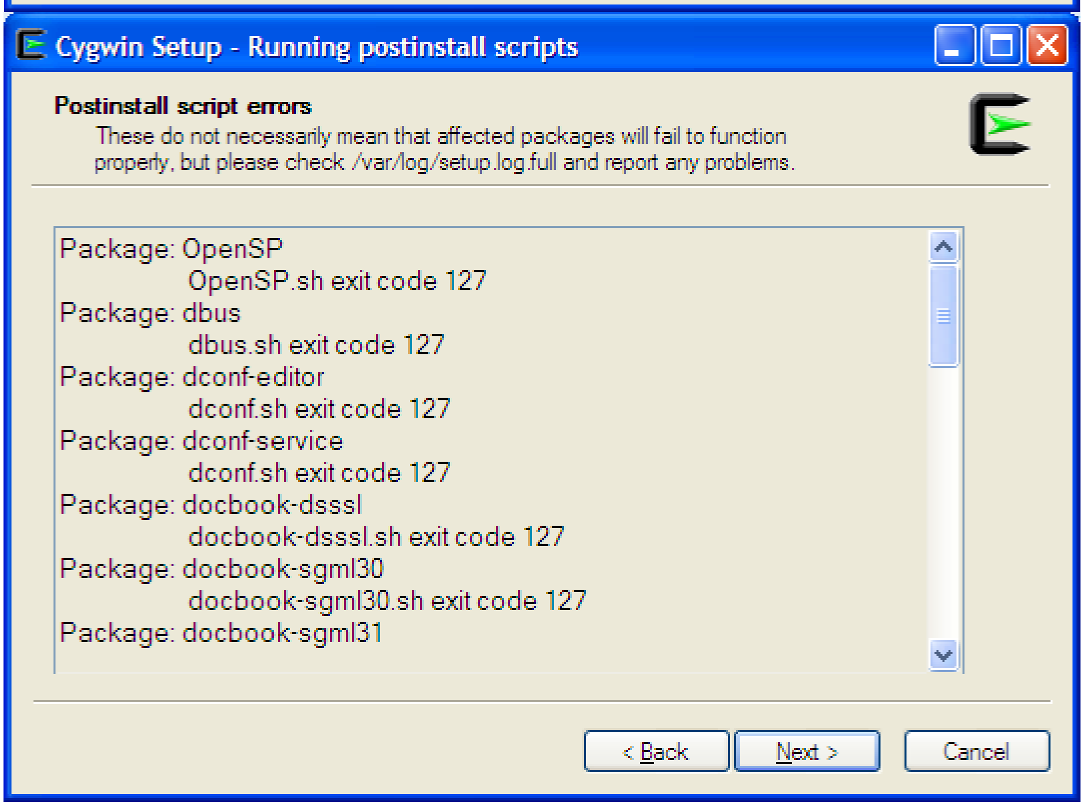

Cygwin Setup
Progress
This page displays the progress of the download or installation.
Checking MD5 for gcc-g++-3.4.4-999
Progress:
Total:
Disk:
< Back
Next >
Cancel

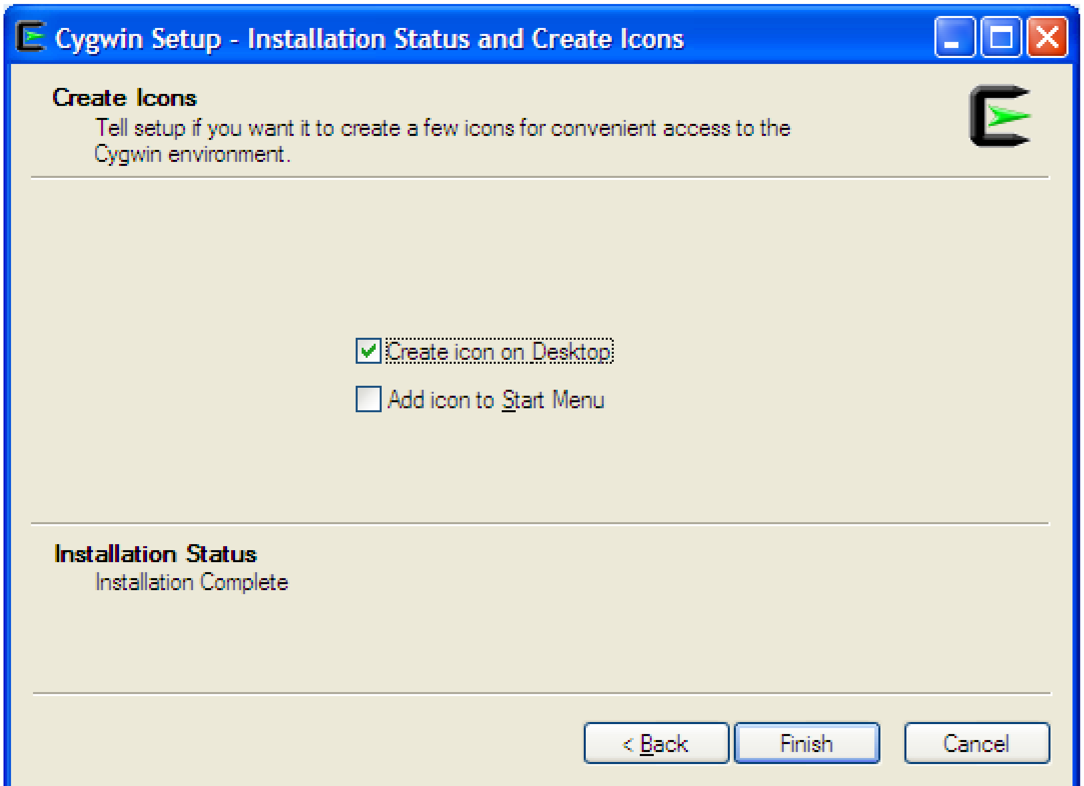

2% - Cygwin Setup
Progress
This page displays the progress of the download or installation.
Installing
R-2.14.2-1
/usr/lib/R/library/spatial/help/spatial.rdx
Progress:
Total:
Disk:
< Back
Next >
Cancel

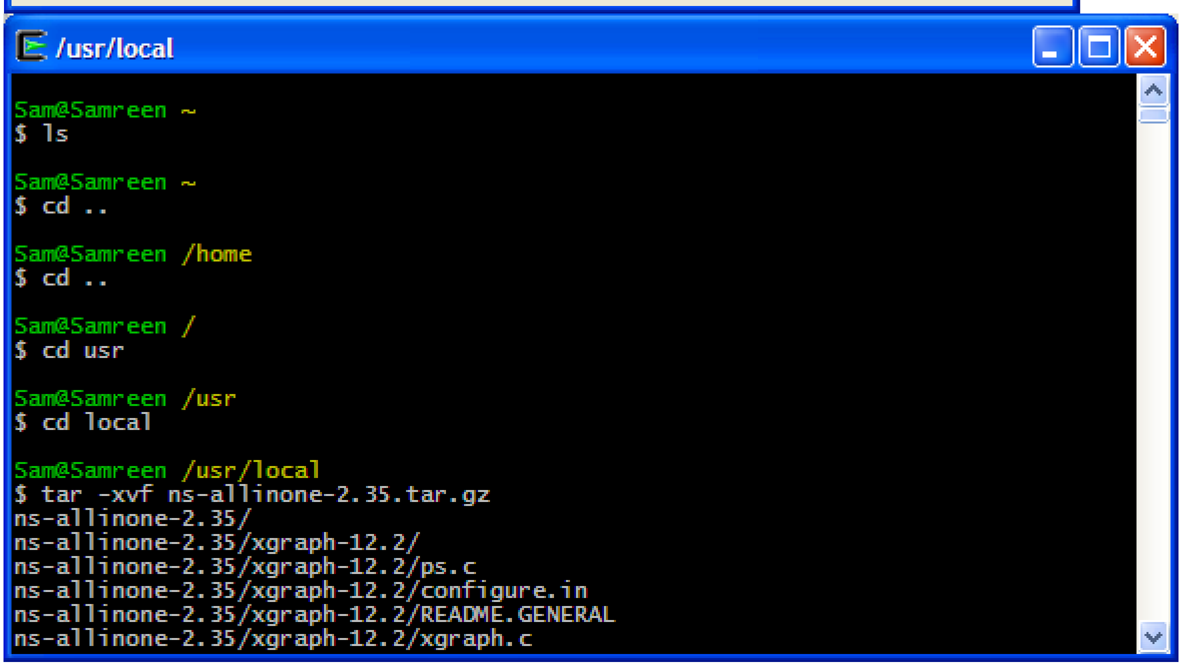

50% - Cygwin Setup
Progress
This page displays the progress of the download or installation.
Installing
mingw-gcc-g++-4.5.2-1
/usr/lib/gcc/i686-pc-mingw32/4.5.2/include/c++/complex.h
Progress:
Total:
Disk:
< Back
Next >
Cancel


74% - Cygwin Setup
Progress
This page displays the progress of the download or installation.
Running...
font-alias
/etc/postinstall/font-alias.sh
Progress:
Total:
Disk:
< Back
Next >
Cancel


99% - Cygwin Setup
Progress
This page displays the progress of the download or installation.
Running...
Unknown package
/etc/postinstall/autorebase.bat
Progress:
Total:
Disk:
< Back
Next >
Cancel


Cygwin Setup - Running postinstall scripts
Postinstall script errors
These do not necessarily mean that affected packages will fail to function properly, but please check /var/log/setup.log.full and report any problems.
Package: OpenSP
OpenSP.sh exit code 127
Package: dbus
dbus.sh exit code 127
Package: dconf-editor
dconf.sh exit code 127
Package: dconf-service
dconf.sh exit code 127
Package: docbook-dsssl
docbook-dsssl.sh exit code 127
Package: docbook-sgml30
docbook-sgml30.sh exit code 127
Package: docbook-sgml31
< Back
Next >
Cancel


Cygwin Setup - Installation Status and Create Icons
Create Icons
Tell setup if you want it to create a few icons for convenient access to the Cygwin environment.
Create icon on Desktop
Add icon to Start Menu
Installation Status
Installation Complete
< Back
Finish
Cancel


/usr/local
Sam@Samreen ~
$ ls
Sam@Samreen ~
$ cd ..
Sam@Samreen /home
$ cd ..
Sam@Samreen /
$ cd usr
Sam@Samreen /usr
$ cd local
Sam@Samreen /usr/local
$ tar -xvf ns-allinone-2.35.tar.gz
ns-allinone-2.35/
ns-allinone-2.35/xgraph-12.2/
ns-allinone-2.35/xgraph-12.2/ps.c
ns-allinone-2.35/xgraph-12.2/configure.in
ns-allinone-2.35/xgraph-12.2/README.GENERAL
ns-allinone-2.35/xgraph-12.2/xgraph.c

/usr/local

```
ns-allinone-2.35/dei80211mr-1.1.4/src/wireless-channelpa.h
ns-allinone-2.35/dei80211mr-1.1.4/src/phymodes.h
ns-allinone-2.35/dei80211mr-1.1.4/src/libinit.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/peerstatsdb_static.h
ns-allinone-2.35/dei80211mr-1.1.4/src/peerstats.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/test_power_profile.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/rbar.h
ns-allinone-2.35/dei80211mr-1.1.4/src/wireless-phymr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/ra-snr.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-802_11mr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/Mac80211EventHandler.h
ns-allinone-2.35/dei80211mr-1.1.4/src/PER.h
ns-allinone-2.35/dei80211mr-1.1.4/src/power_profile.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.h
ns-allinone-2.35/dei80211mr-1.1.4/src/PER.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/papropagation.h
ns-allinone-2.35/dei80211mr-1.1.4/src/per_table_80211b_intersil_HFA3861B.tcl
ns-allinone-2.35/dei80211mr-1.1.4/src/rateadapter.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/ra-snr.cc
ns-allinone-2.35/install

Sam@Samreen /usr/local
$
```

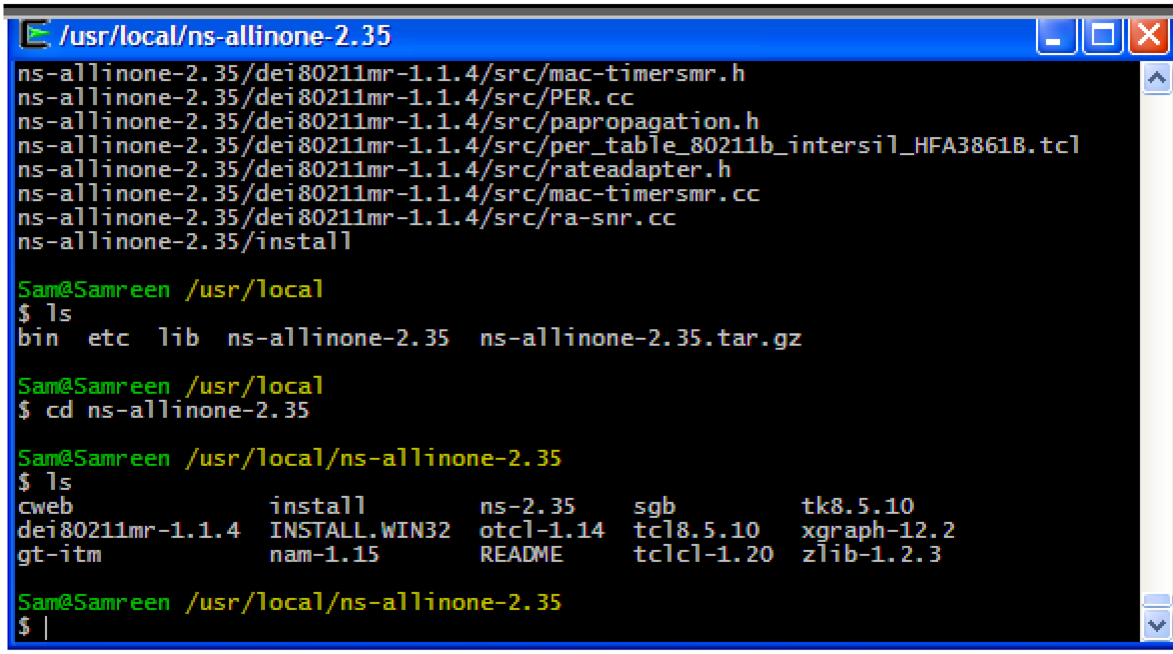
/usr/local/ns-allinone-2.35

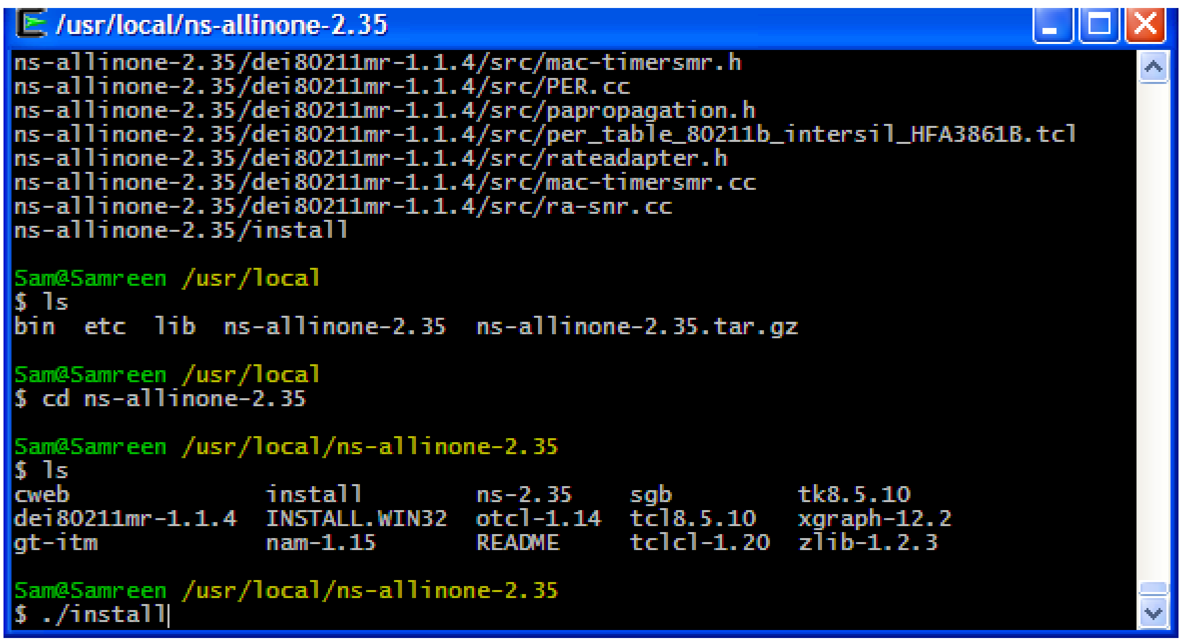
```
ns-allinone-2.35/dei80211mr-1.1.4/src/wireless-phymr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/ra-snr.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-802_11mr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/Mac80211EventHandler.h
ns-allinone-2.35/dei80211mr-1.1.4/src/PER.h
ns-allinone-2.35/dei80211mr-1.1.4/src/power_profile.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.h
ns-allinone-2.35/dei80211mr-1.1.4/src/PER.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/papropagation.h
ns-allinone-2.35/dei80211mr-1.1.4/src/per_table_80211b_intersil_HFA3861B.tcl
ns-allinone-2.35/dei80211mr-1.1.4/src/rateadapter.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/ra-snr.cc
ns-allinone-2.35/install

Sam@Samreen /usr/local
$ ls
bin   etc   lib   ns-allinone-2.35   ns-allinone-2.35.tar.gz

Sam@Samreen /usr/local
$ cd ns-allinone-2.35

Sam@Samreen /usr/local/ns-allinone-2.35
$ |
```


/usr/local/ns-allinone-2.35
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.h
ns-allinone-2.35/dei80211mr-1.1.4/src/PER.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/papropagation.h
ns-allinone-2.35/dei80211mr-1.1.4/src/per_table_80211b_intersil_HFA3861B.tcl
ns-allinone-2.35/dei80211mr-1.1.4/src/rateadapter.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/ra-snr.cc
ns-allinone-2.35/install
Sam@Samreen /usr/local
$ ls
bin etc lib ns-allinone-2.35 ns-allinone-2.35.tar.gz
Sam@Samreen /usr/local
$ cd ns-allinone-2.35
Sam@Samreen /usr/local/ns-allinone-2.35
$ ls
cweb install ns-2.35 sgb tk8.5.10
dei80211mr-1.1.4 INSTALL.WIN32 otcl-1.14 tcl8.5.10 xgraph-12.2
gt-itm nam-1.15 README tclcl-1.20 zlib-1.2.3
Sam@Samreen /usr/local/ns-allinone-2.35
$


/usr/local/ns-allinone-2.35
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.h
ns-allinone-2.35/dei80211mr-1.1.4/src/PER.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/papropagation.h
ns-allinone-2.35/dei80211mr-1.1.4/src/per_table_80211b_intersil_HFA3861B.tcl
ns-allinone-2.35/dei80211mr-1.1.4/src/rateadapter.h
ns-allinone-2.35/dei80211mr-1.1.4/src/mac-timersmr.cc
ns-allinone-2.35/dei80211mr-1.1.4/src/ra-snr.cc
ns-allinone-2.35/install
Sam@Samreen /usr/local
$ ls
bin etc lib ns-allinone-2.35 ns-allinone-2.35.tar.gz
Sam@Samreen /usr/local
$ cd ns-allinone-2.35
Sam@Samreen /usr/local/ns-allinone-2.35
$ ls
cweb install ns-2.35 sgb tk8.5.10
dei80211mr-1.1.4 INSTALL.WIN32 otcl-1.14 tcl8.5.10 xgraph-12.2
gt-itm nam-1.15 README tclcl-1.20 zlib-1.2.3
Sam@Samreen /usr/local/ns-allinone-2.35
$ ./install

Installation and Configuration
Download zip file of ns2.35 (allinone) from:
http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/
Download cygwin setup.exe from www.cygwin.com

Cygwin Installation
1. Click on setup.exe of Cygwin
2. A welcome screen will appear, click next
3. A screen will appear asking for the installation type. Select "Install from Local Directory" and click next.
4. A screen will appear asking for the installation directory where Cygwin will be installed. Select Install for "All users" and click next.
5. A screen will appear asking for the selection of local package directory where downloaded Cygwin files are located. Choose the appropriate directory and click next.
6. A screen will appear asking for the packages to install. Initially the "Default" option is selected. Change it to "Install" and make sure that the following packages are necessarily selected for installation: gcc4, gcc4-g++, gawk, tar, gzip, make, patch, perl, w32api, xorg-server, xinit, libX11-devel, libXmu-devel. Click next.
7. A screen will appear for "Resolving Dependencies". Click next.
8. Next, the Installation will start and the window will show the progress. Wait until the installation is finished. This will take approximately 1 hour or so.
9. After, the installation is finished; a screen will appear with the note "installation completed" and asking for the options of creating desktop icons and start menu entries. Select the desired option and click "Finish".
10. Cygwin is installed now.
11. Open Cygwin through the desktop icon created by the installation.
12. It will show the computer name and the name of person it is registered to in the format "Registered to@Computer name". In my case it is "Sam@Samreen".

NS2 Installation
1. Now, change the directory to usr/local by using the cd command.
2. Copy ns-allinone-2.35.tar.gz to directory C:/cygwin/usr/local
3. Extract the ns2 archive into the usr/local folder.
4. Now change the directory to "ns-allinone-2.35" using the cd command.
5. Type ls to see the list of files in the folder.
6. Type ./install and press enter.
7. The NS-2 installation will start. It will start after making sure the required packages mentioned above are installed. Otherwise it will give error and will ask if the user wants to proceed anyway.
8. Next, the Installation will start and the Console window will show the progress. Wait until the installation is finished. This will also take approximately 1 hour or so.
9. Configure system variables
10. Start Cygwin using the startxwin command
11. Write "ns ex1.tcl" to execute the ns-2 nam interface.