

# ENSC-891 Directed Studies

Summer 2008

ns-BGP Integration with ns-2.33

Will Hrudey  
([whrudey@sfu.ca](mailto:whrudey@sfu.ca))

# Roadmap

- Introduction
- Related work
- Hardware platform
- ns-BGP analysis
- Integration of ns-BGP with ns-2.33
- Validation of ns-2.33-BGP
- Conclusions and future work
- References

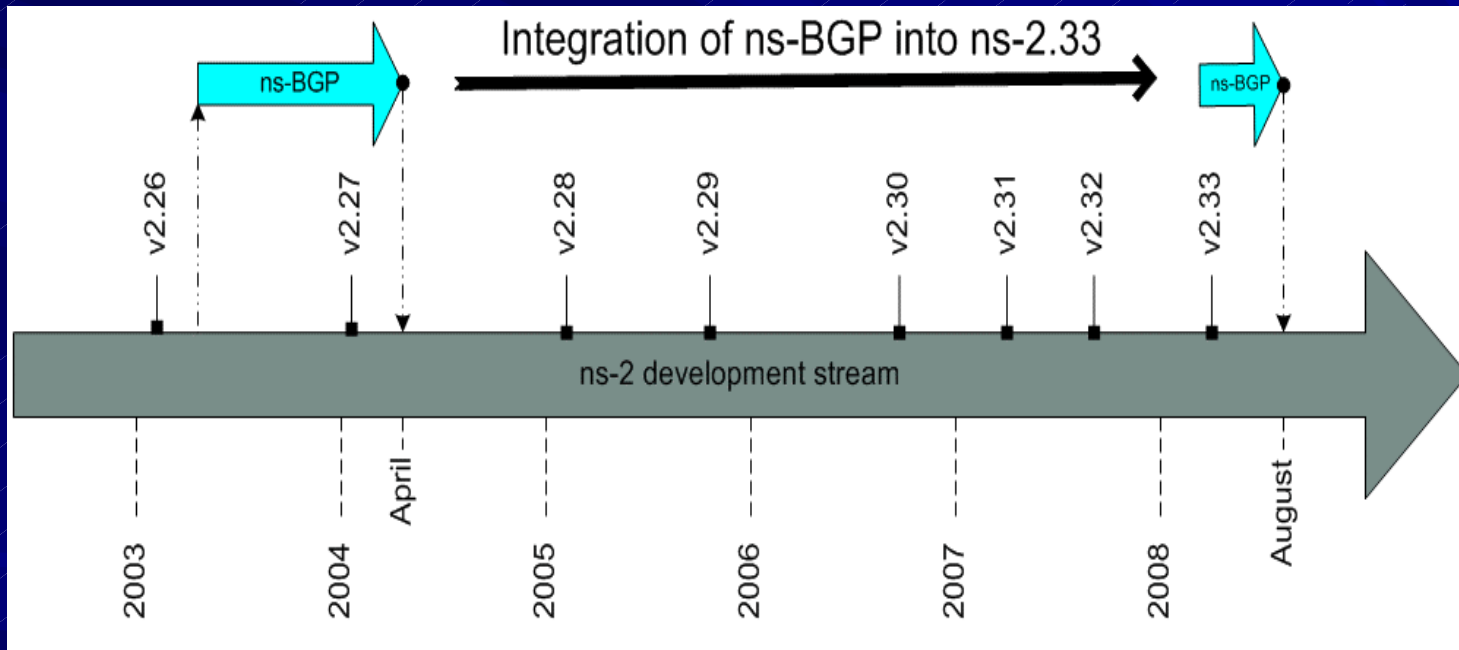
# Project motivation

- BGP performance is affected by the dynamic nature of the Internet<sup>1</sup>
  - simulations can facilitate realistic, flexible BGP routing experimentation
- Aid further BGP research
- ns-2 BGP simulations can enjoy increased simulator reliability, robustness, and feature set
  - updated ns-BGP will benefit from continued ns-2 development and maintenance patches by academic and research communities over the past 5 years

<sup>1</sup> T. D. Feng, R. Ballantyne, and Lj. Trajkovic, "Implementation of BGP in a network simulator," *Applied Telecommunication Symposium, ATS '04*, Arlington, Virginia, Apr. 2004, pp. 149-154.

# Project objective

Integrate the ns-BGP module into the current version of the ns-2 network simulator



# ns-2 overview

## ■ What is ns-2?

- Widely recognized network simulation tool in academic and research communities
- 1989 known as REAL network simulator
- 1995 known as DARPA VINT project
- 200,000 lines of code
- Written in C++, OTcl, Tcl / Tk
- Supported in FreeBSD, Linux, Solaris, Windows, and MAC

# ns-2 overview

- Supports wired and wireless technologies<sup>1,2</sup>
- Provides:
  - routing algorithms
  - transport protocols
  - queuing disciplines
  - traffic sources
  - topology and traffic generators
  - tracing
  - visualization
  - utilities

<sup>1</sup> ns-2 [Online]. Available: <http://www.isi.edu/nsnam/ns> (May 2008).

<sup>2</sup> ns-2 manual [Online]. Available: <http://www.isi.edu/nsnam/ns/doc/index.html> (May 2008).

# BGP overview

## ■ What is BGP?

- 1989 BGP-1 published as RFC 1105
- Current version is BGP-4 published as RFC 1771<sup>1</sup>
- *De facto* inter-domain routing protocol<sup>2</sup>
- Exchanges network layer reachability information (NLRI) between autonomous systems (AS)
- Path vector protocol
- Layered over TCP

<sup>1</sup> Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," RFC 1771, March 1995.

<sup>2</sup> I. Beijnum, *BGP*. Sebastopol, CA: O'Reilly & Associates, 2002.



# BGP overview

- Peer connections established between BGP speakers
- Message exchange driven by routing policies
  - includes connection parameters and routing table updates
- Message types<sup>1,2,3</sup>
  - open, keepalive, notification, notification
- Policies control and modify the routing table
  - determine the conditions for redistributing routes

<sup>1</sup> I. Beijnum, *BGP*. Sebastopol, CA: O'Reilly & Associates, 2002.

<sup>2</sup> Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," RFC 1771, March 1995.

<sup>3</sup> BGP For Internet Service Providers [Online]. Available:  
<http://www.cisco.com/public/cons/seminars/AfNOG3> (June 2008).



# ns-BGP overview

## ■ What is ns-BGP?

- Implements BGP-4 in ns-2<sup>1</sup>
- BGP module ported from SSFNet
- IPv4 addressing, TCP sockets, packet forwarding added
- ns-BGP node derived from ns-2 unicast node
- Routing achieved through forwarding and control planes:
  - forwarding plane classifies and forwards packets
  - control plane handles route creation, computation, routing algorithms, routing table management

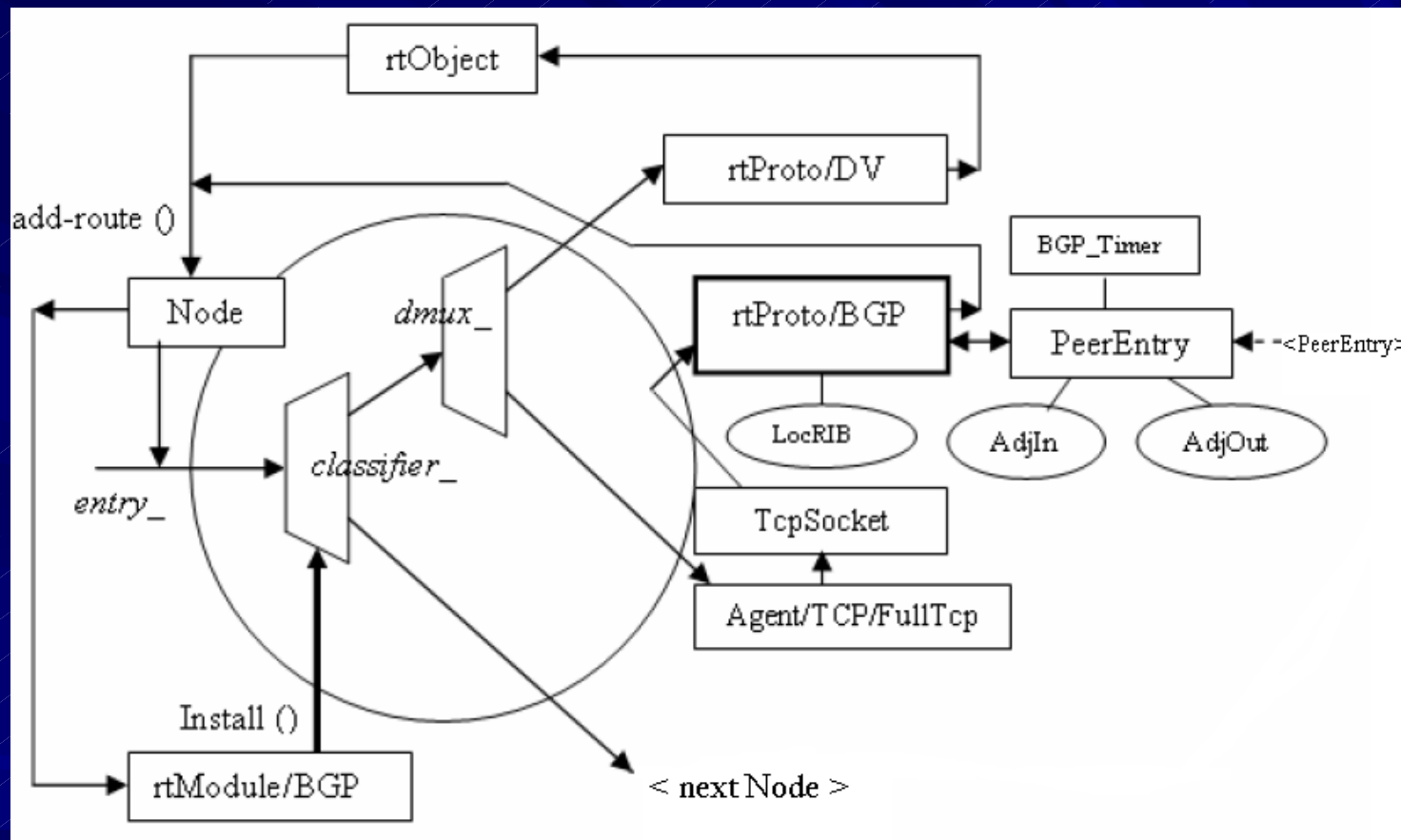
<sup>1</sup> T. D. Feng, R. Ballantyne, and Lj. Trajkovic, "Implementation of BGP in a network simulator," *Applied Telecommunication Symposium, ATS '04*, Arlington, Virginia, Apr. 2004, pp. 149-154.

# ns-BGP overview

- Four primary classes used in implementation<sup>1</sup>
  - TcpSocket : UNIX-like socket programming interface
  - IPv4Classifier: classifies incoming packets
  - rtModule/BGP: replaces existing routing module
  - rtProto/BGP: implements BPG-4 using dual classes

<sup>1</sup> T. D. Feng, R. Ballantyne, and Lj. Trajkovic, "Implementation of BGP in a network simulator," *Applied Telecommunication Symposium, ATS '04*, Arlington, Virginia, Apr. 2004, pp. 149-154.

# ns-BGP unicast node<sup>1</sup>



<sup>1</sup> T. D. Feng, R. Ballantyne, and Lj. Trajkovic, "Implementation of BGP in a network simulator," *Applied Telecommunication Symposium, ATS '04*, Arlington, Virginia, Apr. 2004, pp. 149-154.

# ns-BGP overview

- Optional features included:
  - multiple exit discriminator
  - aggregator
  - community
  - originator ID
  - cluster list path attributes
  - route reflections
- Experimental features included:
  - sender side loop detection
  - withdrawal, per-peer, per-destination rate limiting
  - unjittered minimum route advertisement timer

# Roadmap

- Introduction
- **Related work**
- Hardware platform
- ns-BGP analysis
- Integration of ns-BGP with ns-2.33
- Validation of ns-2.33-BGP
- Conclusions and future work
- References

# Related work

- OPNET BGP [Online]. Available: <http://www.opnet.com> (June 2008).
  - Implementation greatly differs from ns-2
- SSFNet [Online]. Available: <http://www.ssfnet.org/homePage.html> (May 2008).
  - Simulation tool not as widely adopted as ns-2
  - ns-BGP was ported from SSFNet and adapted accordingly
- C-BGP [Online]. Available: <http://cbgp.info.ucl.ac.be/wiki/index.php> (June 2008).
  - Dedicated BGP solver rather than flexible network simulator
- GNU Zebra BGP daemon [Online]. Available: <http://www.zebra.org/zebra/BGP.html#BGP> (June 2008).
  - Ported to ns-2 around same time as ns-BGP
  - Written in C
- BGP++ [Online]. Available: <http://www.ece.gatech.edu/research/labs/MANIACS/BGP++> (June 2008).
  - Written for ns-2 and GTNetS simulators
  - Actually a port of Zebra BGP adapted to C++ environment



# Roadmap

- Introduction
- Related work
- **Hardware platform**
- ns-BGP analysis
- Integration of ns-BGP with ns-2.33
- Validation of ns-2.33-BGP
- Conclusions and future work
- References



# Hardware platform

- Project kickoff:
  - Toshiba Tecra S2 laptop
  - Intel Pentium M processor / 1GB RAM
  - Windows XP service pack 2 (host OS)
  
- Hardware replaced at 75% project completion point:
  - Dell D630 laptop
  - Intel duo core T7250 2.0 GHz processor / 4GB RAM
  - Vista Business Edition service pack 1 (host OS)

# Multiple environments

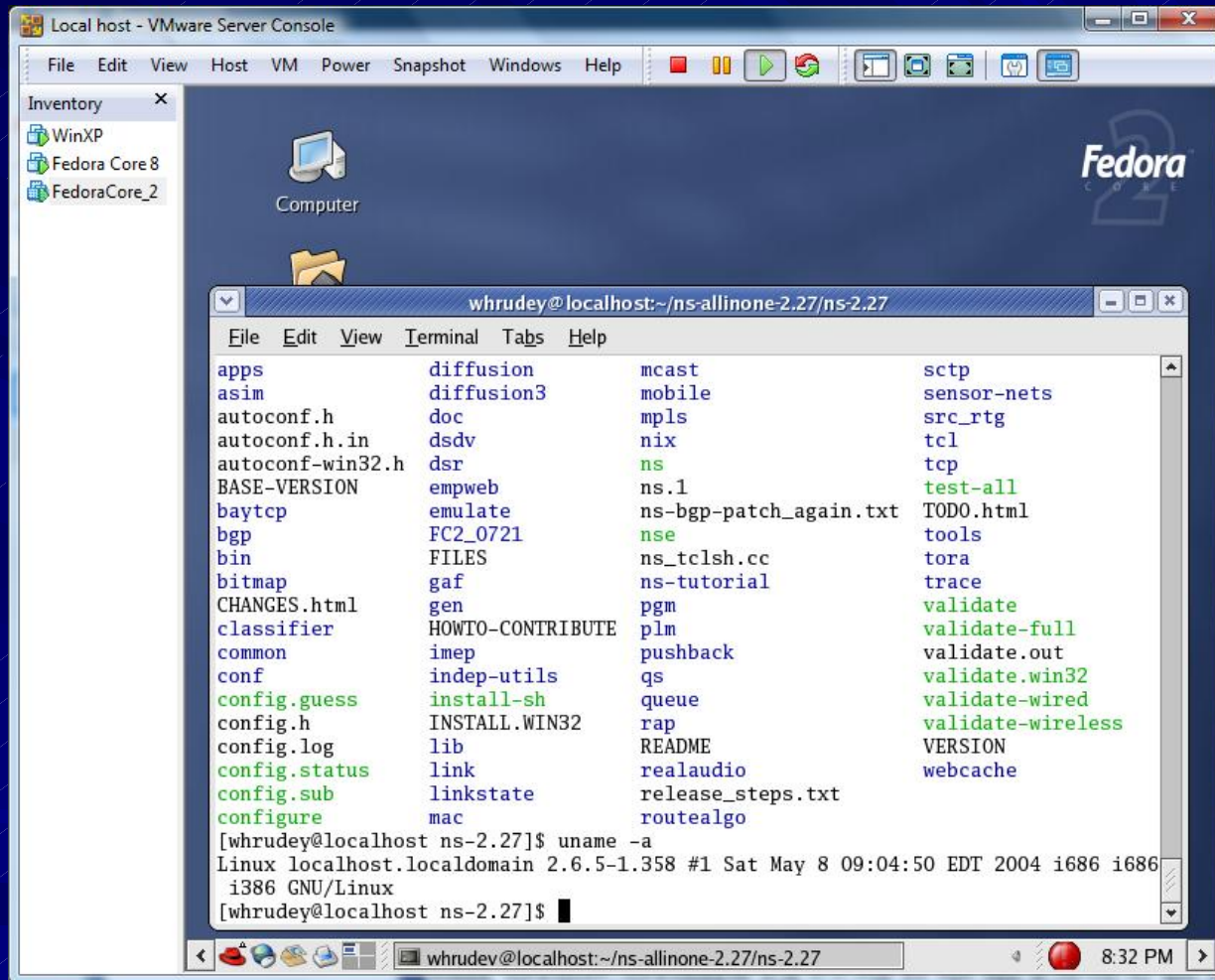
- Virtual machines (VM):
  - VMware Server 1.0.6
  - Accommodate multiple guest OS's (i.e. Linux)
  - Target integration environment:
    - current Fedora Core (FC) distribution
    - current ns-2 release
  - Native ns-BGP development environment:
    - most recent Fedora distribution able to run ns-2.27
    - iterative, systematic “downgrading” of FC distribution version

# Derived configurations

Derived configurations	Software components									
	Vista Business Edition	VMWare Server 1.06	WinXP SP3 / Cygwin	Fedora Core 2	Fedora Core 4	Fedora Core 8	ns-2.26	ns-2.27	ns-2.33	ns-BGP 2.0 (original release)
Test-1	▪	▪	▪				▫	▫	▪	▫
Test-2 ( target integration environment )	▪	▪				▪	▫	▫	▪	▫
Test-3	▪	▪			▪		▫	▫		
Test-4 ( native ns-BGP development environment )	▪	▪		▪			▪	▪		▪

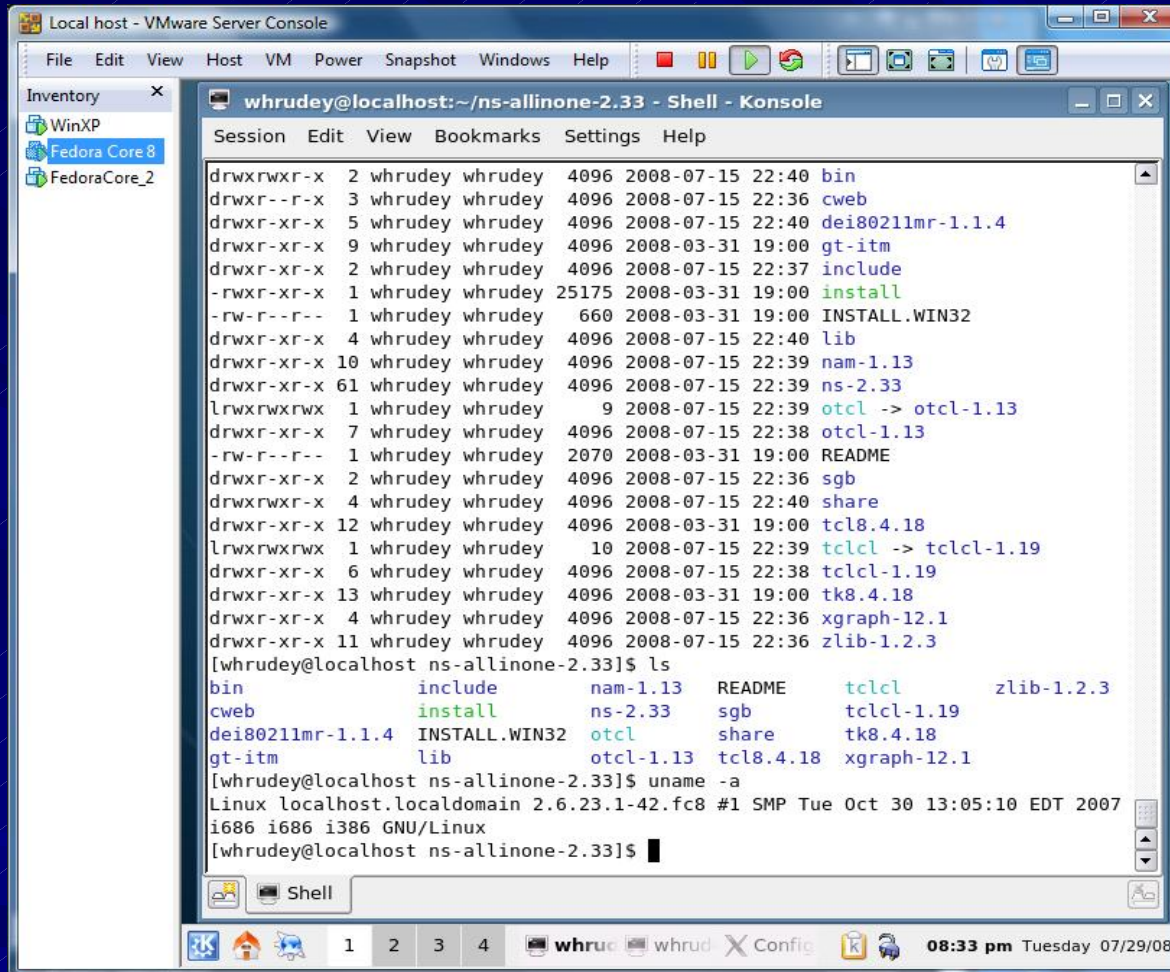
- denotes the successful inclusion of a given software component
- denotes the unsuccessful inclusion of a given software component

# Fedora Core 2 virtual machine



ns-2.27 session running in Fedora Core 2 VM within VMware Server

# Fedora Core 8 virtual machine



```
Local host - VMware Server Console
File Edit View Host VM Power Snapshot Windows Help
Inventory
WinXP
Fedora Core 8
FedoraCore_2

whrudey@localhost:~/ns-allinone-2.33 - Shell - Konsole
Session Edit View Bookmarks Settings Help
drwxrwxr-x 2 whrudey whrudey 4096 2008-07-15 22:40 bin
drwxr--r-x 3 whrudey whrudey 4096 2008-07-15 22:36 cweb
drwxr-xr-x 5 whrudey whrudey 4096 2008-07-15 22:40 dei80211mr-1.1.4
drwxr-xr-x 9 whrudey whrudey 4096 2008-03-31 19:00 gt-itm
drwxr-xr-x 2 whrudey whrudey 4096 2008-07-15 22:37 include
-rwxr-xr-x 1 whrudey whrudey 25175 2008-03-31 19:00 install
-rw-r--r-- 1 whrudey whrudey 660 2008-03-31 19:00 INSTALL.WIN32
drwxr-xr-x 4 whrudey whrudey 4096 2008-07-15 22:40 lib
drwxr-xr-x 10 whrudey whrudey 4096 2008-07-15 22:39 nam-1.13
drwxr-xr-x 61 whrudey whrudey 4096 2008-07-15 22:39 ns-2.33
lrwxrwxrwx 1 whrudey whrudey 9 2008-07-15 22:39 otcl -> otcl-1.13
drwxr-xr-x 7 whrudey whrudey 4096 2008-07-15 22:38 otcl-1.13
-rw-r--r-- 1 whrudey whrudey 2070 2008-03-31 19:00 README
drwxr-xr-x 2 whrudey whrudey 4096 2008-07-15 22:36 sgb
drwxrwxr-x 4 whrudey whrudey 4096 2008-07-15 22:40 share
drwxr-xr-x 12 whrudey whrudey 4096 2008-03-31 19:00 tcl8.4.18
lrwxrwxrwx 1 whrudey whrudey 10 2008-07-15 22:39 tclcl -> tclcl-1.19
drwxr-xr-x 6 whrudey whrudey 4096 2008-07-15 22:38 tclcl-1.19
drwxr-xr-x 13 whrudey whrudey 4096 2008-03-31 19:00 tk8.4.18
drwxr-xr-x 4 whrudey whrudey 4096 2008-07-15 22:36 xgraph-12.1
drwxr-xr-x 11 whrudey whrudey 4096 2008-07-15 22:36 zlib-1.2.3
[whrudey@localhost ns-allinone-2.33]$ ls
bin          include      nam-1.13    README      tclcl       zlib-1.2.3
cweb        install      ns-2.33     sgb         tclcl-1.19
dei80211mr-1.1.4  INSTALL.WIN32  otcl       share       tk8.4.18
gt-itm      lib          otcl-1.13  tcl8.4.18  xgraph-12.1
[whrudey@localhost ns-allinone-2.33]$ uname -a
Linux localhost.localdomain 2.6.23.1-42.fc8 #1 SMP Tue Oct 30 13:05:10 EDT 2007
i686 i686 i386 GNU/Linux
[whrudey@localhost ns-allinone-2.33]$
```

ns-2.33 session running in Fedora Core 8 VM within VMware Server



# Roadmap

- Introduction
- Related work
- Hardware platform
- **ns-BGP analysis**
- Integration of ns-BGP with ns-2.33
- Validation of ns-2.33-BGP
- Conclusions and future work
- References

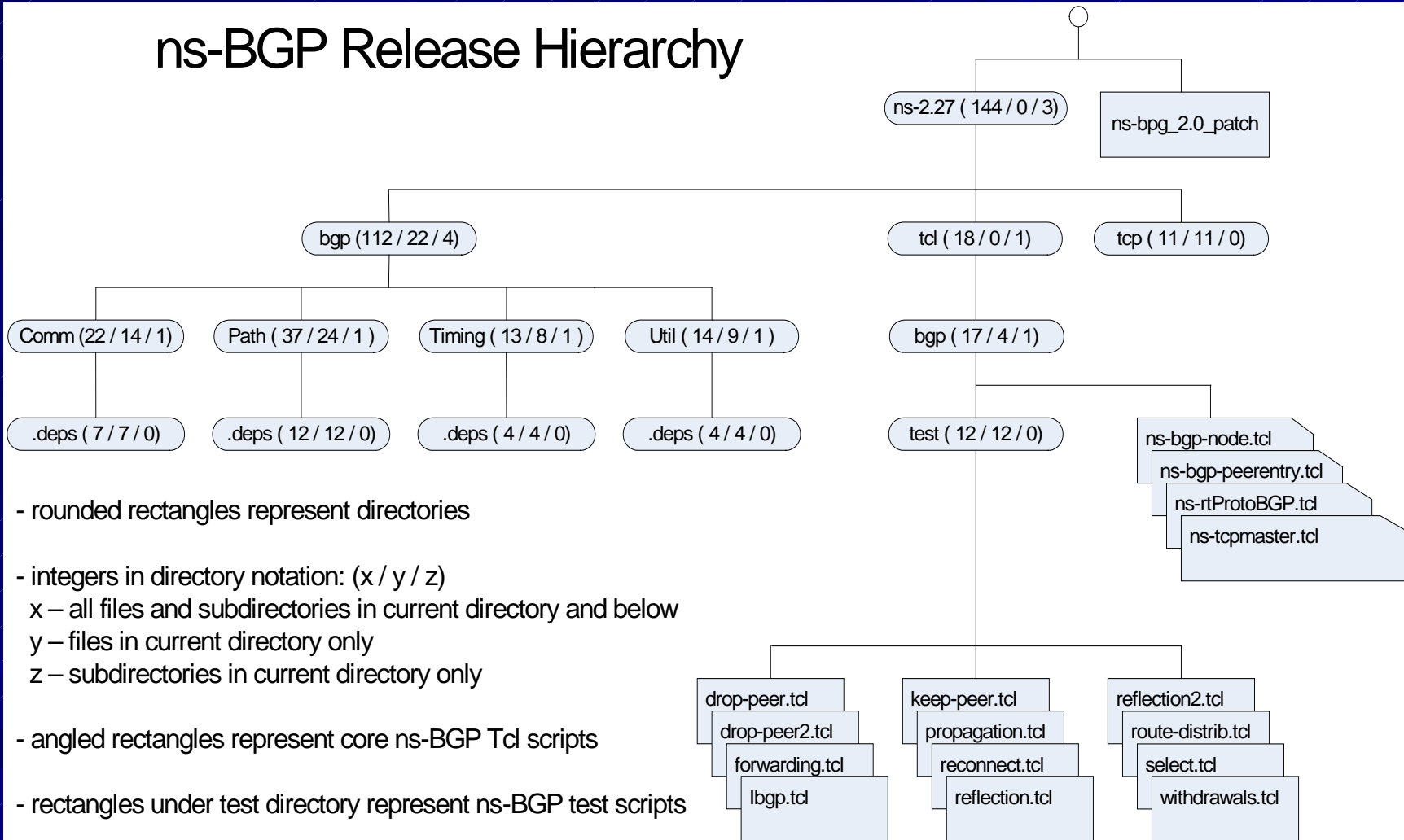
# ns-BGP release details

- Software release format
  - 110 KB compressed tar ball / 1.64 MB uncompressed
- 145 files in release:
  - 104 source files:
    - 46 C++ header files (.h)
    - 41 C++ code files (.cc)
    - 16 Tcl files (.tcl)
    - patch file (contains edits to 16 core ns-2 files)
  - 27 dependency files (.Po)
  - 13 subdirectories
  - 1 readme file (.txt)



# ns-BGP analysis

## ns-BGP Release Hierarchy



- rounded rectangles represent directories
- integers in directory notation: (x / y / z)  
 x – all files and subdirectories in current directory and below  
 y – files in current directory only  
 z – subdirectories in current directory only
- angled rectangles represent core ns-BGP Tcl scripts
- rectangles under test directory represent ns-BGP test scripts

# Roadmap

- Introduction
- Related work
- Hardware platform
- ns-BGP analysis
- **Integration of ns-BGP with ns-2.33**
- Validation of ns-2.33-BGP
- Conclusions and future work
- References

# Integration strategy

## ■ Objectives:

- Integrate the ns-BGP module, which was designed and developed for ns-2.27, into the current version of simulator: ns-2.33
- Retain all subsequent ns-2 enhancements and maintenance updates that have occurred since ns-BGP was released

# Two stage integration approach

- Code integration:
  - migrate new ns-BGP source files (non-overlapping) to target ns-2.33 directories
  - integrate logic changes in patch file pertaining to existing core ns-2.27 source files to ns-2.33
- Compilation modifications
  - resolve dependencies tied to native development compiler

# Code integration details

- File types to migrate to ns-2.33:
  - C++ code files (.cc)
  - C++ header files (.h)
  - Tcl script files (.tcl)
  - Readme.txt file
- Propagate patch file code edits to core ns-2.33 files
- File types that don't require migration
  - Dependency files (.Po)

# Summary of modified source files

Files	Changes	
	Merge	Compilation
ns-2.33/common/node.cc	o	
ns-2.33/common/node.h	o	
ns-2.33/common/packet.h	X	
ns-2.33/common/simulator.cc	X	
ns-2.33/Makefile.in	o	
ns-2.33/routing/route.cc	o	
ns-2.33/routing/rmodule.cc	o	
ns-2.33/routing/rmodule.h	o	
ns-2.33/tcl/lib/ns-default.tcl	o	
ns-2.33/tcl/lib/ns-lib.tcl	o	
ns-2.33/tcl/lib/ns-node.tcl	X	
ns-2.33/tcp/rq.cc	o	
ns-2.33/tcp/rq.h	o	
ns-2.33/tcp/scoreboard-rq.cc	o	
ns-2.33/tcp/tcp-full.cc	X	
ns-2.33/tcp/tcp-full.h	X	
ns-2.33/bgp/Util/ipaddress.cc		X
ns-2.33/bgp/Util/ipaddress.h		X
ns-2.33/tcp/send_queue.cc		X

Files listed in the merge column reflect core ns-2 files to be patched with logic edits detailed in ns-BGP patch file

Files listed in compilation column reflect files that required modification to compile successfully

o denotes basic complexity  
X denotes moderate complexity

# Code integration example

- Enumerated packet types in ns-2.27 were changed by ns-2 designers to unsigned integers in ns-2.33 to allow dynamic packet types:

- ns-2.27 packet.h "code snippet"

```
enum packet_t {  
    PT_RTPROTO_BGP,  
    PT_TCPMASTER,  
    PT_PEERENTRY,  
};
```

- ns-2.33 packet.h "code snippet"

```
typedef unsigned int packet_t;  
static const packet_t PT_RTPROTO_BGP = 70;  
static const packet_t PT_TCPMASTER = 71;  
static const packet_t PT_PEERENTRY = 72;
```



# Compilation error example

## ■ Standard Template Library - list container error

```
tcp/send_queue.cc: In member function `TcpData*  
SendQueue::get_data(int, int)`:  
tcp/send_queue.cc:57: error: conversion from `int` to non-  
scalar type `std::_List_iterator <SendData>' requested  
tcp/send_queue.cc:71: error: no match for `operator==` in  
`targetIterator == 0`  
/usr/lib/gcc/i386-redhat-linux/4.1.2/../../../../include/c++/4.1.2/bits/stl_list.h:169:  
note: candidates are: bool  
std::_List_iterator<_Tp>::operator==(const  
std::_List_iterator <_Tp>&) const [with _Tp = SendData]  
make: *** [tcp/send_queue.o] Error 1
```

## ■ Resolution

```
57: list<SendData>::iterator targetIterator= (list<SendData>::iterator) NULL;  
71: if(targetIterator == (list<SendData>::iterator) NULL)
```

# ns-BGP limitations

- Functionality is that of original ns-BGP implementation
- Any original ns-BGP software bugs and/or computational inefficiencies will still exist
- Any subsequent ns-BGP enhancements by academic and research communities not included

# Roadmap

- Introduction
- Related work
- Hardware platform
- ns-BGP analysis
- Integration of ns-BGP with ns-2.33
- **Validation of ns-2.33-BGP**
- Conclusions and future work
- References

# Two stage validation approach

- Compilation phase
  - code syntax validated by successful compilation
- ns-BGP test scripts:
  - BGP-4 compliant tests
  - comparison of each ns-BGP test script's standard output and trace output across ns-2.27 and ns-2.33 environments

# Validation results

## ■ ns-BGP test scripts

Test Scripts	ns-2.27		ns-2.33		Equivalence	
	Output filename	Output file size	Output filename	Output filename size	File	Standard Out
drop-peer.tcl	drop-peer.nam	19055	drop-peer.nam	19055	✓	✓
drop-peer2.tcl	drop-peer2.nam	37213	drop-peer2.nam	37213	✓	✓
forwarding.tcl	forwarding.nam	3210797	forwarding.nam	3210797	✓	✓
	forwarding.out	1498410	forwarding.out	1498410	✓	✓
ibgp.tcl	ibgp.nam	24528	ibgp.nam	24528	✓	✓
keep-peer.tcl	keep-peer.nam	23892	keep-peer.nam	23892	✓	✓
propagation.tcl	propagation.nam	19779	propagation.nam	19779	✓	✓
reconnect.tcl	reconnect.nam	39793	reconnect.nam	39793	✓	✓
reflection.tcl	reflection.nam	87614	reflection.nam	87614	✓	✓
reflection2.tcl	reflection2.nam	85405	reflection2.nam	85405	✓	✓
route-distrib.tcl	route-distrib.nam	11668	route-distrib.nam	11668	✓	✓
select.tcl	select.nam	31338	select.nam	31338	✓	✓
withdrawals.tcl	withdrawals.nam	15060	withdrawals.nam	15060	✓	✓

File equivalence indicates both file size (in bytes) and file contents match  
Standard out equivalence indicates script execution output to terminal match

# Roadmap

- Introduction
- Related work
- Hardware platform
- ns-BGP analysis
- Integration of ns-BGP with ns-2.33
- Validation of ns-2.33-BGP
- **Conclusions and future work**
- References

# Conclusions

- Project objective achieved
- ns-BGP integration challenges overcome:
  - Code merge ambiguities and complexities
  - Resolution to lengthy compilation errors
- Validation demonstrates integrity and equivalence:
  - ns-BGP trace output and standard output identical between ns-2.27 and ns-2.33
  - Within the given time constraints, validation provides a high degree of confidence in ns-BGP release integrity
- Integrated ns-2.33 release remains stable
  - No observed core dumps



# Future work

## ■ Challenges:

- Learning curve with ns-2 internals, ns-BGP, C++, OTcl
- Deriving ns-2.27 native development environment

## ■ Future Work:

- Add policy routing
- Add route flap damping<sup>1</sup>
- Add adaptive minimal route advertisement interval<sup>2</sup>

<sup>1</sup> W. Shen and Lj. Trajkovic, "BGP route flap damping algorithms," *Proc. SPECTS 2005*, Philadelphia, PA, July 2005, pp. 488-495.

<sup>2</sup> N. Laskovic and Lj. Trajkovic, "BGP with an adaptive minimal route advertisement interval," *Proc. 25th IEEE Int. Performance, Computing, and Communications Conference*, Phoenix, AZ, April 2006, pp. 135-142.

# References

- T. D. Feng, R. Ballantyne, and Lj. Trajkovic, "Implementation of BGP in a network simulator," *Applied Telecommunication Symposium, ATS '04*, Arlington, Virginia, Apr. 2004, pp. 149-154.
- I. Beijnum, *BGP*. Sebastopol, CA: O'Reilly & Associates, 2002.
- ns-2 [Online]. Available: <http://www.isi.edu/nsnam/ns> (May 2008).
- ns-2 manual [Online]. Available: <http://www.isi.edu/nsnam/ns/doc/index.html> (May 2008).
- Y. Rekhter and T. Li, "*A border gateway protocol 4 (BGP-4)*," RFC 1771, March 1995.
- R. Johnsonbaugh and J. Kalin, *Object-Oriented Programming in C++*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- B. Welch, K. Jones, and J. Hobbs, *Practical Programming in Tcl and Tk 4/e*. Prentice Hall, 2003.
- BGP For Internet Service Providers [Online]. Available: <http://www.cisco.com/public/cons/seminars/AfNOG3> (June 2008).

# References

- SSFNet [Online]. Available: <http://www.ssfnet.org/homePage.html> (May 2008).
- OPNET BGP [Online]. Available: <http://www.opnet.com> (June 2008).
- C-BGP [Online]. Available: <http://cbgp.info.ucl.ac.be/wiki/index.php> (June 2008).
- GNU Zebra BGP daemon [Online]. Available: <http://www.zebra.org/zebra/BGP.html#BGP> (June 2008).
- BGP++ [Online]. Available: <http://www.ece.gatech.edu/research/labs/MANIACS/BGP++> (June 2008).
- W. Shen and Lj. Trajkovic, "BGP route flap damping algorithms," *Proc. SPECTS 2005*, Philadelphia, PA, July 2005, pp. 488-495.
- N. Laskovic and Lj. Trajkovic, "[BGP with an adaptive minimal route advertisement interval](#)," *Proc. 25th IEEE Int. Performance, Computing, and Communications Conference*, Phoenix, AZ, April 2006, pp. 135-142.