NS-2 Trace Formats

More NS2 information.

This document lists various trace formats used by the NS-2 Network Simulator. The information is based on NS2 version 2.1b9a. I've done my best to document correctly all of the trace formats I've come across, but be warned that this list is not complete, and may contain errors. If you find any errors or omissions, please let me know. The trace formats are:

- Normal trace formats
- Wireless trace formats:
    - Old wireless trace formats
    - New wireless trace formats
    - AODV routing protocol trace formats
    - DSDV routing protocol trace formats
    - DSR routing protocol trace formats
    - TORA routing protocol trace formats
    - Mobile node movement and energy trace formats
- NAM (Network Animator) trace formats
- Change Log

The various traces begin with a single character or abbreviation that indicates the type of trace, followed by a fixed or variable trace format. The tables listing the trace formats differ between fixed and variable trace formats:

- For fixed trace formats, the table lists the event the triggers the trace under the Event heading and the characters that start the trace under the Abbreviation heading. The format is listed across the last two columns, and the the type and value for each element of the format are listed beneath under the Type and Value headings. Some events have multiple trace formats.

- For variable trace formats, the table lists the event the triggers the trace under the Event heading and the characters that start the trace under the Abbreviation heading. The last three columns list the possible flags, types, and values for the event under the Flag, Type, and Value headings.

The tables that list the additional wireless trace information do not have an Abbreviation column, since the information is appended to the end of the regular wireless trace format.

The following people sent feedback and questions that helped improve this page. Thanks for the help!

- Peter Liscovius
- George Kinal
- Daniel Brenner
- Qingjiang Tian
- Tahir Mahjabeen

Normal trace formats

This information comes from The ns Manual "Trace and Monitoring Support: Trace File Format" chapter. This trace is used normal wired operations. The trace starts with one of four possible characters.

Event	Abbreviation	Type	Value
Normal Event	r: Receive d: Drop e: Error +: Enqueue -: Dequeue	%g %d %d %s %d %s %d %d.%d %d.%d %d %d	
		double	Time
		int	Source Node
		int	Destination Node
		string	Packet Name
		int	Packet Size
		string	Flags
		int	Flow ID
		int	Source Address

		int	Destination Address
		int	Sequence Number
		int	Unique Packet ID

Depending on the packet type, the trace may log additional information:

Event	Type	Value
TCP Trace	%d 0x%x %d %d	
	int	Ack Number
	hexadecimal	Flags
	int	Header Length
	int	Socket Address Length
Satellite Trace	%.2f %.2f %.2f %.2f	
	double	Source Latitude
	double	Source Longitude
	double	Destination Latitude
	double	Destination Longitude

Wireless Trace Formats

This section covers the various wireless trace format:

- Old wireless trace format
- New wireless trace format
- AODV routing protocol
- DSDV routing protocol
- DSR routing protocol

- TORA routing protocol
- Mobile node movement and energy trace formats

Old Wireless Trace Formats

This information comes from The ns Manual "Mobile Networking in ns: Trace Support" chapter, and the "trace/cmu-trace.cc" file. Wireless traces begin with one of four characters followed by one of two different trace formats, depending on whether the trace logs the X and Y coordinates of the mobile node.

Event	Abbreviation	Type	Value
Wireless Event	s: Send r: Receive d: Drop f: Forward	%.9f %d (%6.2f %6.2f) %3s %4s %d %s %d [%x %x %x %x]	
		%.9f _%d_ %3s %4s %d %s %d [%x %x %x %x]	
		double	Time
		int	Node ID
		double	X Coordinate (If Logging Position)
		double	Y Coordinate (If Logging Position)
		string	Trace Name
		string	Reason
		int	Event Identifier
		string	Packet Type
		int	Packet Size
		hexadecimal	Time To Send Data
		hexadecimal	Destination MAC Address
		hexadecimal	Source MAC Address
		hexadecimal	Type (ARP, IP)

Some older versions of NS2 (such as 2.1b5) have five hexidecimal values between the square braces. The first hexidecimal value is the MAC frame control information, and the remaining hexidecimal values are the same as listed above.

Depending on the packet type, the trace may log additional information:

Event	Type	Value

ARP Trace	------- [%s %d/%d %d/%d]	
	string	Request or Reply
	int	Source MAC Address
	int	Source Address
	int	Destination MAC Address
	int	Destination Address
DSR Trace	%d [%d %d] [%d %d %d %d->%d] [%d %d %d %d->%d]	
	int	Number Of Nodes Traversed
	int	Routing Request Flag
	int	Route Request Sequence Number
	int	Routing Reply Flag
	int	Route Request Sequence Number
	int	Reply Length
	int	Source Of Source Routing
	int	Destination Of Source Routing
	int	Error Report Flag (?)
	int	Number Of Errors
	int	Report To Whom
	int	Link Error From
	int	Link Error To
AODV Trace	[0x%x %d %d [%d %d] [%d %d]] (REQUEST)	
	hexadecimal	Type
	int	Hop Count
	int	Broadcast ID
	int	Destination
	int	Destination Sequence Number
	int	Source
	int	Source Sequence Number
	[0x%x %d [%d %d] %f] (%s)	
	hexadecimal	Type
	int	Hop Count

	int	Destination
	int	Destination Sequence Number
	double	Lifetime
	string	Operation (REPLY, ERROR, HELLO)
TORA Trace	[0x%x %d] (QUERY)	
	hexadecimal	Type
	int	Destination
	0x%x %d (%f %d %d %d %d) (UPDATE)	
	hexadecimal	Type
	int	Destination
	double	Tau
	int	Oid
	int	R
	int	Delta
	int	ID
	[0x%x %d %f %d] (CLEAR)	
	hexadecimal	Type
	int	Destination
	double	Tau
	int	Oid
IP Trace	------- [%d:%d %d:%d %d %d]	
	int	Source IP Address
	int	Source Port Number
	int	Destination IP Address
	int	Destination Port Number
	int	TTL Value
	int	Next Hop Address, If Any
TCP Trace	[%d %d] %d %d	
	int	Sequence Number
	int	Acknowledgment Number
	int	Number Of Times Packet Was Forwarded
	int	Optimal Number Of Forwards

CBR Trace	[%d] %d %d	
	int	Sequence Number
	int	Number Of Times Packet Was Forwarded
	int	Optimal Number Of Forwards
IMEP Trace	[%c %c %c 0x%04x]	
	char	Acknowledgment Flag
	char	Hello Flag
	char	Object Flag
	hexadecimal	Length
RCA Trace (from MIT Leach code)	------- [%c %d %d %d]	
	char	Operation (A, R, D)
	int	RCA Source
	int	RCA Link Destination
	int	RCA MAC Destination

New Wireless Trace Formats

This information comes from The ns Manual "Mobile Networking in ns: Revised format for wireless traces" chapter, and the "trace/cmu-trace.cc" file. Similar to the old format, in the new format wireless traces begin with one of four characters. This is followed by flag/value pairs similar to NAM traces. The first letter of flags with two letters designates the flag type:

- N: Node Property
- I: IP Level Packet Information
- H: Next Hop Information
- M: MAC Level Packet Information
- P: Packet Specific Information

Event	Abbreviation	Flag	Type	Value
		-t	double	Time (* For Global Setting)
		-Ni	int	Node ID
		-Nx	double	Node X Coordinate
		-Ny	double	Node Y Coordinate

Wireless Event	s: Send r: Receive d: Drop f: Forward	-Nz	double	Node Z Coordinate
		-Ne	double	Node Energy Level
		-Nl	string	Network trace Level (AGT, RTR, MAC, etc.)
		-Nw	string	Drop Reason
		-Hs	int	Hop source node ID
		-Hd	int	Hop destination Node ID, -1, -2
		-Ma	hexadecimal	Duration
		-Ms	hexadecimal	Source Ethernet Address
		-Md	hexadecimal	Destination Ethernet Address
		-Mt	hexadecimal	Ethernet Type
		-P	string	Packet Type (arp, dsr, imep, tora, etc.)
		-Pn	string	Packet Type (cbr, tcp)

Note that the value for the -Hd flag may be -1 or -2. -1 means that the packet is a broadcast packet, and -2 means that the destination node has not been set. -2 is typically seen for packets that are passed between the agent (-Nl AGT) and routing (-Nl RTR) levels.

Depending on the packet type, the following flags may be used:

Event	Flag	Type	Value
ARP Trace	-Po	string	Request or Reply
	-Pms	int	Source MAC Address
	-Ps	int	Source Address
	-Pmd	int	Destination MAC Address
	-Pd	int	Destination Address
	-Ph	int	Number Of Nodes Traversed
	-Pq	int	Routing Request Flag
	-Ps	int	Route Request Sequence Number

DSR Trace	-Pp	int	Routing Reply Flag
	-Pn	int	Route Request Sequence Number
	-Pl	int	Reply Length
	-Pe	int->int	Source->Destination Of Source Routing
	-Pw	int	Error Report Flag (?)
	-Pm	int	Number Of Errors
	-Pc	int	Report To Whom
	-Pb	int->int	Link Error From Link A to Link B
AODV Trace	-Pt	hexadecimal	Type
	-Ph	int	Hop Count
	-Pb	int	Broadcast ID
	-Pd	int	Destination
	-Pds	int	Destination Sequence Number
	-Ps	int	Source
	-Pss	int	Source Sequence Number
	-Pl	double	Lifetime
	-Pc	string	Operation (REQUEST, REPLY, ERROR, HELLO)
TORA Trace	-Pt	hexadecimal	Type
	-Pd	int	Destination
	-Pa	double	Time
	-Po	int	Creator ID
	-Pr	int	R
	-Pe	int	Delta
	-Pi	int	ID
	-Pc	string	Operation (QUERY, UPDATE, CLEAR)
IP Trace	-Is	int.int	Source Address And Port
	-Id	int.int	Destination Address And Port
	-It	string	Packet Type
	-Il	int	Packet Size
	-If	int	Flow ID
	-Ii	int	Unique ID

	-Iv	int	TTL Value
TCP Trace	-Ps	int	Sequence Number
	-Pa	int	Acknowledgment Number
	-Pf	int	Number Of Times Packet Was Forwarded
	-Po	int	Optimal Number Of Forwards
CBR Trace	-Pi	int	Sequence Number
	-Pf	int	Number Of Times Packet Was Forwarded
	-Po	int	Optimal Number Of Forwards
IMEP Trace	-Pa	char	Acknowledgment Flag
	-Ph	char	Hello Flag
	-Po	char	Object Flag
	-Pl	hexadecimal	Length

AODV Trace Formats

AODV traces begin with an "A", followed by the AODV trace. This information comes from the "aodv/aodv_logs.cc" source file.

Event	Abbreviation	Type	Value
Delete Link	A	%.9f _%d_ deleting LL hop to %d (delete %d is %s)	
		double	Time
		int	Index
		int	Destination
		int	Deleted Link Count
		string	Link State (VALID, INVALID)
Broken Link	A	%.9f _%d_ LL unable to deliver packet %d to %d (%d) (reason = %d, ifqlen = %d)	
		double	Time
		int	Index
		int	Unique Packet ID
		int	Next Hop
		int	Broken Link Count
		int	Transmit Reason

		int	Queue Length
Keeping Bad Link	A	%.9f _%d_ keeping LL hop to %d (keep %d is %s)	
		double	Time
		int	Index
		int	Destination
		int	Kept Bad Link Count
		string	Link State (VALID, INVALID)

DSDV Trace Formats

DSDV traces begin with a "V", followed by additional characters to indicate the exact DSDV trace. This information comes from the "dsdv/dsdv.cc" source file.

Event	Abbreviation	Type	Value
Trace Packet	VPU VTU	%.5f _%d_ [%d] (%d,%d,%d) ...	
		double	Time
		int	Reporting Address
		int	Count
		int	Destination
		int	Distance (Metric)
		int	Sequence Number
Periodic Callback	VPC	%.5f _%d_	
		double	Time
		int	Reporting Address
Timeout	VTO	%.5f _%d_ %d->%d	
		double	Time
		int	Reporting Address
		int	Reporting Address (Should be Source???)
		int	Routing Table Destination
		%.5f _%d_ marking %d	
		double	Time
		int	Reporting Address

		int	Routing Table Destination
Lost Link	VLL	%.8f %d->%d lost at %d	
		double	Time
		int	Source
		int	Destination
		int	Reporting Address
Lost Packet	VLP	%.5f %d:%d->%d:%d lost at %d [hop %d]	
		double	Time
		int	Source
		int	Source Port
		int	Destination
		int	Destination Port
		int	Reporting Address
		int	Routing Table Destination
Change Table	VCT	%.5f _%d_ %d	
		double	Time
		int	Reporting Address
		int	Routing Table Destination
Weighted Settling Time	VWST	%.12lf frm %d to %d wst %.12lf nxthp %d [of %d]	
		double	Time
		int	Reporting Address
		int	Routing Table Destination
		double	Weighted Settling Time
		int	Next Hop
		int	Distance (Metric)
Update Route	VSD VSU	%.5f _%d_ (%d,%d->%d,%d->%d,%d->%d,%f)	
		double	Time
		int	Reporting Address
		int	Old Destination
		int	Old Distance (Metric) or -1
		int	New Distance (Metric)

		int	Old Sequence Number or -1
		int	New Sequence Number
		int	Old Hop or -1
		int	New Hop
		double	When Okay To Advertise This Route
Queue Packet	VBP	%.5f _%d_ %d:%d -> %d:%d	
		double	Time
		int	Reporting Address
		int	Source Address
		int	Source Port
		int	Destination Address
		int	Destination Port
Routing Packets Outside Domain	VFP	%.5f _%d_ %d:%d -> %d:%d	
		double	Time
		int	Reporting Address
		int	Source Address
		int	Source Port
		int	Destination Address
		int	Destination Port
Table Dump	VTD	%.5f %d:%d	
		double	Time
		int	Reporting Address
		int	Source Address
		int	Source Port

DSR Trace Formats

DSR traces begin with an "S", which may be followed by additional characters to indicate the exact DSR trace. Each trace has one or more formats. This information comes from the "dsr/dsragent.cc", "dsr/linkcache.cc", "dsr/mobicache.cc", "dsr/routecache.cc", and "dsr/simplecache.cc" source files.

Event	Abbreviation	Type	Value
		%.9f _%s_ originating %s -> %s	

Send	S	double	Time
		string	ID
		string	Source
		string	Destination
Have A Route	S$hit	%.5f _%s_ %s -> %s %s	
		double	Time
		string	ID
		string	Source
		string	Destination
		string	Route
Don't Have A Route	S$miss	%.5f _%s_ %s -> %s	
		double	Time
		string	ID
		string	ID (Should be Source???)
		string	Destination
Configuration parameters. All strings are either "on" or "off"	Sconfig	%.5f tap: %s snoop: rts? %s errs? %s	
		double	Time
		string	Use TAP
		string	Snoop Source Routes
		string	Snoop Forwarded Errors
		%.5f salvage: %s !bd replies? %s	
		double	Time
		string	Salvage With Cache
		string	Don't Salvage Bad Replies
		%.5f grat error: %s grat reply: %s	
		double	Time
		string	Propagate Last Error
		string	Send Grat Replies
		%.5f $reply for props: %s ring 0 search: %s	
		double	Time
		string	Reply From Cache On Propagating
		string	Ring Zero Search

		%.5f using MOBICACHE	
		double	Time
		%.5f using LINKCACHE	
		double	Time
		%.5f _%s_ stuck into send buff %s -> %s	
		double	Time
		string	ID
		string	Source
		string	Destination
		%.5f _%s_ checking for route for dst %s	
		double	Time
		string	ID
		string	Destination
		%.5f _%s_ sendbuf pkt to %s liberated by handlePktWOSR	
		double	Time
		string	ID
		string	Destination
		%.9f _%s_ splitting %s to %s	
		double	Time
		string	ID
		string	Route
		string	Route Copy
		%.9f _%s_ liberated from sendbuf %s->%s %s	
		double	Time
		string	ID
		string	Source
		string	Destination
		string	Route
		%.5f _%s_ unwrapping nested route error	
		double	Time
		string	ID

Debug message	Sdebug	_%s_ tap saw error %d
		string: ID
		int: Header UID
		_%s_ tap saw route reply %d %s
		string: ID
		int: Header UID
		string: Reply Path
		_%s_ tap saw route use %d %s
		string: ID
		int: Header UID
		string: Route
		%s consider grat arp for %s
		string: ID
		string: Route
		_%s_ not bothering to send route error to ourselves
		string: ID
		%.5f _%s_ sending into dead-link (nest %d) tell %d %d -> %d
		double: Time
		string: ID
		int: Number Of Route Errors
		int: Report To Address
		int: From Address
		int: To Address
		%.9f _%s_ adding %s [%d %.9f] ...
		double: Time
		string: ID
		string: Path
		int: Link Type
		double: Time Added
		%.9f _%s_ checking %s [%d %.9f] ...

		double	Time
		string	ID
		string	Path
		int	Link Type
		double	Time Added
		%.9f _%s_ freshening %s->%s to %d %.9f	
		double	Time
		string	ID
		string	Path
		string	Next Path
		int	Link Type
		double	Time Added
Errors	SDFU	%.5f _%s_ dumping maximally nested error %s %d -> %d	
		double	Time
		string	ID
		string	Tell ID
		int	From
		int	To
		ran off the end of a source route	
		non route containing packet given to acceptRouteReply	
		route error beyond end of source route????	
		route error forwarding route request????	
Flowstate	SFs	%.9f _%s_ %d [%s -> %s] %d(%d) to %d	
		double	Time
		string	ID
		int	Header UID
		string	Source
		string	Destination
		int	Flow ID
		int	Flow Header

		int	Next Hop
Established Flowstate	SFESTs	%.9f _%s_ %d [%s -> %s] %d(%d) to %d %s	
		double	Time
		string	ID
		int	Header UID
		string	Source
		string	Destination
		int	Flow ID
		int	Flow Header
		int	Next Hop
		string	Addresses
Flow ARS	SFARS	%.9f _%s_ %d [%s -> %s] %d %d	
		double	Time
		string	ID
		int	Header ID
		string	Source
		string	Destination
		int	Flow ID
		int	Amt
Flow Error	SFEr	%.9f _%s_ from %d re %d : %d [%d]	
		double	Time
		string	ID
		int	Source
		int	Flow Destination
		int	Flow ID (-1 For Default)
		int	Count (-1 For No Flow Index)
Unknown Flow	SFErr	%.5f _%s_ %d -> %d : %d	
		double	Time
		string	ID
		int	Source
		int	Flow Destination
		int	Flow ID

Flow Forward	SFf	%.9f _%s_ %d [%s -> %s] %d to %d	
		double	Time
		string	ID
		int	Header ID
		string	Source
		string	Destination
		int	Flow ID
		int	Next Hop
Interface Queue	SIFQ	%.5f _%s_ len %d	
		double	Time
		string	ID
		int	Queue Length
Send Out Packet With Route	SO	%.9f _%s_ originating %s %s	
		double	Time
		string	ID
		string	Protocol Name
		string	Route
		%.9f _%s_ cache-summary %d %d %d %d | %d %.9f %d %d | %d %d %d %d %d | %d %d %d %d %d | %d %d %d %d %d %d %.9f	
		%.9f _%s_ cache-summary %d %d %d %d | %d %.9f %d %d | %d %d %d %d %d | %d %d %d %d %d | %d %d %d %d %d %d	
		double	Time
		string	ID
		int	Route Count
		int	Route Bad Count
		int	Subroute Count
		int	Subroute Bad Count
		int	Link Bad Count
		double	Average Bad Time Per Link
		int	Link Bad Tested
		int	Link Good Tested

Route Cache - Summary	SRC	int	Route Add Count
		int	Route Add Bad Count
		int	Subroute Add Count
		int	Subroute Add Bad Count
		int	Link Add Tested
		int	Route Notice Count
		int	Route Notice Bad Count
		int	Subroute Notice Count
		int	Subroute Notice Bad Count
		int	Link Notice Tested
		int	Route Find Count
		int	Route Find For Me
		int	Route Find Bad Count
		int	Route Find Miss Count
		int	Subroute Find Count
		int	Subroute Find Bad Count
		double	Link Good Time (Only In First Format)
Route Cache - Node Cache Dump (only with patch)	SRC	%.9f _%s_ cache-dump p %d %d | %d %d %d %s | ... | s %d %d | %d %d %d %s | ... |	
		double	Time
		int	Source Node
		int	Primary Cache Current Size
		int	Primary Cache Maximum Size
		int	Cache Entry Index
		int	Cache Entry Length
		int	Cache Entry Address
		string	Cache Entry Dump
		int	Secondary Cache Current Size
		int	Secondary Cache Maximum Size
		int	Cache Entry Index
		int	Cache Entry Length

		int	Cache Entry Address
		string	Cache Entry Dump
Route Cache - Node Cache Dump, No Primary Cache (only with patch)	SRC	%.9f _%s_ cache-dump s %d %d | %d %d %d %s | ... |	
		double	Time
		int	Source Node
		int	Secondary Cache Current Size
		int	Secondary Cache Maximum Size
		int	Cache Entry Index
		int	Cache Entry Length
		int	Cache Entry Address
		string	Cache Entry Dump
Route Cache - Find Route Cache Hit	SRC	%.9f _%s_ $hit for %s in %s %s	
		double	Time
		string	ID
		string	Destination
		string	Primary Or Secondary Cache
		string	Route
Route Cache - Find Route Cache Miss	SRC	%.9f _%s_ find-route [%d] %s->%s miss %d %.9f	
		double	Time
		string	ID
		int	Hardcoded Zero
		string	ID (Should Be Source???)
		string	Destination
		int	Hardcoded Zero
		double	Hardcoded Zero
Route Cache - New Route Contains Cached Route	SRC	%.9f _%s_ %s suffix-rule (len %d/%d) %s	
		double	Time
		string	ID
		string	Cache Name (primary, secondary)
		int	Path Length

		int	Route Length
		string	Route Dump
Route Cache - New Route Contained In Cache	SRC	%.9f _%s_ %s prefix-rule (len %d/%d) %s	
		double	Time
		string	ID
		string	Cache Name (primary, secondary)
		int	Path Length
		int	Route Length
		string	Route Dump
Route Cache - Discard Route	SRC	%.9f _%s_ %s evicting %s	
		double	Time
		string	ID
		string	Name
		string	Dumped Route
Route Cache - Discard Route	SRC	%.9f _%s_ evicting %d %d %s	
		double	Time
		string	ID
		int	Route Length -1
		int	Number Of Bad Routes
		string	Name
Route Cache - Add Route After Dumping Route	SRC	%.9f _%s_ while adding %s	
		double	Time
		string	ID
		string	Name
		string	Added Path
Route Cache - Truncating Route To Remove Dead Link	SRC	%.9f _%s_ %s truncating %s %s	
		double	Time
		string	ID
		string	Name
		string	Route
		string	Owner
		%.9f _%s_ to %s %s	

Route Cache - Truncated Or Removed Route With Dead Link	SRC	double	Time
		string	ID
		string	Route
		string	Owner
Route Cache - Dead Link	SRC	%.9f _%s_ dead link %s->%s	
		double	Time
		string	ID
		string	From
		string	To
Route Cache - Dead Link	SRC	%.9f _%s_ %s [%d %d] %s->%s dead %d %.9f	
		double	Time
		string	ID
		string	Operation In Progress (add-route, notice-route, find-route, dead-link, evicting-route, check-cache)
		int	Route Length
		int	Route Index
		string	Route
		string	Next Route
		int	Link Type
		double	Time Added
Route Cache - Resurrected Link	SRC	%.9f _%s_ resurrected-link [%d %d] %s->%s dead %d %.9f	
		double	Time
		string	ID
		int	Route Length
		int	Route Index
		string	Route
		string	Next Route
		int	Link Type
		double	Time Added
		%.9f _%s_ adding rt %s from %s	

Route Cache - Add Route	SRC	double	Time
		string	ID
		string	Route
		string	From
Route Cache - Dump Dijkstra	SRC	%.9f _%s_ dijkstra *%d* %d,%d,%d ...	
		double	Time
		string	ID
		int	Destination
		int	Index
		int	Estimated Shortest Path To Vertex (d)
		int	Predecessors For Vertex (pi)
Route Cache - Dump Link	SRC	%.9f _%s_ dump-link %d->%d, ...	
		double	Time
		string	ID
		int	Index
		int	Link Destination
Route Cache - Cache Expire Bits	SRC	%.9f _%s_ cache-expire-bits %d %d %d %d	
		double	Time
		string	ID
		int	Expire Stats 0
		int	Expire Stats 1
		int	Expire Stats 2
		int	Expire Stats 3
		%.5f _%s_ dropped %s #%d (ignored)	
		double	Time
		string	ID
		string	Source
		int	Route Request Sequence
		%.9f _%s_ discarding %s #%d (ifq length %d)	
		double	Time
		string	ID
		string	Source

int	Route Request Sequence
int	Queue Length
%.9f _%s_ discarding %s #%d (free air time %f)	
double	Time
string	ID
string	Source
int	Route Request Sequence
int	Free Air Time
%.5f _%s_ dropped %s #%d (prop limit exceeded)	
double	Time
string	ID
string	Source
int	Route Request Sequence
%.5f _%s_ dropped %s #%d (SR full)	
double	Time
string	ID
string	Source
int	Route Request Sequence
%.5f _%s_ rebroadcast %s #%d ->%s %s	
double	Time
string	ID
string	Source
int	Route Request Sequence
string	Destination
string	Route
%.9f _%s_ cache-reply-sent %s -> %s #%d (len %d) %s	
double	Time
string	ID
string	Source

Route Request/Reply	SRR	string	Destination
		int	Request Sequence Number
		int	Route Length
		string	Route
		%.5f _%s_ RR-not-sent %s -> %s	
		double	Time
		string	ID
		string	Route Request Source
		string	Route Request Destination
		%.5f _%s_ new-request %d %s #%d -> %s	
		double	Time
		string	ID
		int	Maximum Propagation
		string	Source
		int	Route Request Sequence
		string	Destination
		%.9f _%s_ reply-sent %s -> %s #%d (len %d) %s	
		double	Time
		string	ID
		string	Source
		string	Destination
		int	Route Request Sequence
		int	Route Length
		string	Route
		%.9f _%s_ reply-received %d from %s %s #%d -> %s %s	
		double	Time
		string	ID
		int	Good Reply (0, 1)
		string	Source
		string	First Reply Route

		int	Route Request Sequence
		string	Last Reply Route
		string	Reply Route
		%.9f _%s_ dead-link tell %d %d -> %d	
		double	Time
		string	ID
		int	Report To Address
		int	From Address
		int	To Address
		%.9f _%s_ gratuitous-reply-sent %s -> %s (len %d) %s	
		double	Time
		string	ID
		string	Source
		string	Destination
		int	Route Length
		string	Route
		%.5f _%s_ --- %d dropping bad-reply %s -> %s	
		double	Time
		string	ID
		int	Header UID
		string	Source
		string	Destination
		%.5f _%s_ salvaging %s -> %s --- %d with %s	
		double	Time
		string	ID
		string	Source
		string	Destination
		int	Header UID
		string	Route
		%.5f _%s_ adding to SB --- %d %s -> %s [%d]	

Salvage	Ssalv	double	Time
		string	ID
		int	Header UID
		string	Source
		string	Destination
		int	Salvaged
		%.5f _%s_ dropping --- %d %s -> %s [%d]	
		double	Time
		string	ID
		int	Header UID
		string	Source
		string	Destination
		int	Salvaged
Packet dropped by send buffer in DSR agent	Ssb	%.5f _%s_ dropped %s -> %s	
		double	Time
		string	ID
		string	Source
		string	Destination
Send Failure	SSendFailure	%.9f _%s_ %d %d %d:%d %d:%d %s->%s %d %d %d %d %s	
		double	Time
		string	ID
		int	Header UID
		int	Protocol Type
		int	Source
		int	Source Port
		int	Destination
		int	Destination Port
		string	From ID
		string	To ID
		int	Hops: From -> To
		int	Hops: Source -> Destination

		int	Hops: From -> Destination
		int	Number Of Addresses
		string	Header
Transmit Failed	SxmitFailed	%.5f _%s_ %d->%d god okays #%d	
		double	Time
		string	ID
		int	From ID
		int	To ID
		int	Number of Wrong Link Errors
Maximally Nested Flow Error	SYFU	%.5f _%s_ dumping maximally nested Flow error %d -> %d	
		double	Time
		string	ID
		int	Source
		int	Flow Destination
Attempted To Add Bad Route To Cache		%.9f _%s_ adding bad route to cache %s %s	
		double	Time
		string	ID
		string	Source
		string	Route

TORA Trace Formats

TORA traces begin with an "T", followed by one of several formats. This information comes from the "tora/tora.cc", "tora/tora_api.cc", and "tora/tora_io.cc" source files.

Event	Abbreviation	Type	Value
		%.9f _%d_ tora sendQRY %d	
		double	Time
		int	Address
		int	ID
		%.9f _%d_ QRY %d for %d (rtreq set)	
		double	Time

TORA Event	T	int	Address
		int	TORA Destination Index
		int	Index
		%.9f _%d_ tora enq %d->%d	
		double	Time
		int	Address
		int	Source
		int	Destination
		%.9f _%d_ received `UPD` from non-neighbor %d	
		double	Time
		int	Address
		int	Source
		%.9f _%d_ received `CLR` from non-neighbor %d	
		double	Time
		int	Address
		int	Source

Mobile node movement and energy trace formats

Mobile node traces begin with "M" or "N" This information comes from the "common/mobilenode.cc" source file.

Event	Abbreviation	Type	Value
Mobile Node Movement	M	%.5f %d (%.2f, %.2f, %.2f), (%.2f, %.2f), %.2f	
		double	Time
		int	Address (Node ID?)
		double	X Coordinate
		double	Y Coordinate
		double	Z Coordinate
		double	Destination X Coordinate
		double	Destination Y Coordinate

		double	Movement Speed
Mobile Node Energy	N	-t %f -n %d -e %f	
		double	Time
		int	Address (Node ID?)
		double	Energy

- Mobilenode movement and energy trace formats

NAM Trace Formats

The general format for a NAM trace is a single letter abbreviation followed by one or more flag/value pairs. This information comes from The ns Manual "Nam Trace" chapter. It can also be generated by running "nam -p". Note that all flags may not be used every time.

Event	Abbreviation	Flag	Type	Value
Comment -- this line is ignored	#			
Dummy event to be used in time synchronization	T	-t	time	Time
Node	n	-t	time	Time
		-s	int	Node ID
		-u	double	X Velocity
		-U	double	X Velocity
		-V	double	Y Velocity
		-v	shape	Shape (circle, box, hexagon)
		-c	color	Color
		-z	double	Size Of Node
		-a	int	Address
		-x	double	X Location
		-y	double	Y Location
		-Z	double	Z Location (Not Supported)
		-i	color	Label Color
		-b	string	Label

		-l	string	Label
		-o	color	Previous Color
		-S	string	State (UP, DOWN, COLOR)
		-L	string	Previous Label
		-p	string	Label Location
		-P	string	Previous Label Location
		-i	color	Inside Label Color
		-I	color	Previous Inside Label Color
		-e	color	Label Color
		-E	color	Previous Label Color
		-T	double	Duration Of Movement
		-w	flag	Wireless Node
Link	l	-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-r	double	Transmission Rate
		-D	double	Delay
		-h	double	Length
		-O	orientation	Orientation
		-b	string	Label
		-c	color	Color
		-o	color	Previous Color
		-S	string	State (UP, DOWN)
		-l	string l	Label
		-L	string	Previous Label
		-e	color	Label Color
		-E	color	Previous Label Color
		-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-e	int	Extent
		-a	int	Packet Color Attribute ID

Packet	h: Hop r: Receive d: Drop Line +: Enqueue -: Dequeue	-i	int	ID
		-l	int	Energy
		-c	string	Conversation
		-x	comment	Comment
		-p	string	Packet Type
		-k	string	Packet Type
		-y	comment	
		-S	int	
		-m	int	
		-f	int	
Session	E: Enqueue D: Dequeue P: Drop	-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-e	int	Extent
		-a	int	Attribute
		-i	int	ID
		-l	int	Energy
		-c	string	Conversation
		-x	comment	Comment
		-p	string	Packet Type
		-k	string	Packet Type
Agent	a	-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-x	flag	Remove Agent
		-n	string	Agent Name
Feature	f	-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-x	flag	Remove Feature
		-T	char	Type
		-n	string	Name

		-a	string	Agent
		-v	string	Value
		-o	string	Previous Value
Group	G	-t	time	Time
		-n	string	Name
		-i	int	Node ID
		-a	int	Group ID
		-x	flag	Remove From Group
Lan link	L	-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-o	orientation	Orientation
		-O	orientation	Orientation
Mark node	m	-t	time	Time
		-n	string	Name
		-s	int	Node ID
		-c	string	Color
		-h	string	Shape (circle, square, hexagon)
		-X	flag	Remove Mark
Routing event	R	-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-g	int	Multicast Group
		-p	packet source	Packet Source ID Or *
		-n	flag	Negative Cache
		-x	flag	This Route Timed Out
		-T	double	Timeout
		-m	string	Mode (IIF Or OIF)
Execute tcl expression	v	-t	time	Time
		-e	tcl expression	Tcl Script
Set trace file version	V	-t	time	Time
		-v	string	Version

		-a	int	Attribute
Use nam graph	N			
Wireless range	W	-t	time	Time
		-x	int	X
		-y	int	Y
Energy status -- for future use	g	-t	time	Time
Hierarchical address space configuration -- initialization only	A	-t	time	Time
		-n	int	Hierarchy
		-p	int	Port Shift
		-o	hexadecimal	Port Mask
		-c	int	Multicast Shift
		-a	int	Multicast Mask
		-h	int	Hierarchy
		-m	int	Node Shift
		-s	int	Node Mask
Color table configuration -- initialization only	c	-t	time	Time
		-i	int	ID
		-n	string	Color
Create packet queue -- initialization only	q	-t	time	Time
		-s	int	Source ID
		-d	int	Destination ID
		-a	orientation	Orientation
Layout lan	X	-t	time	Time
		-n	string	Name
		-r	double	Rate
		-D	double	Delay
		-o	orientation	Orientation
		-O	orientation	Orientation

For Packet events (entries starting with "h", "r", "d", "+", or "-"), the comment field (field after "-x" has the following format:

Event	Type	Value
Node Trace	{%s.%s %s.%s %d %s %s}	
	string	Source Node Address
	string	Source Node Port
	string	Destination Node Address
	string	Destination Node Port
	int	Sequence Number
	string	Flags
	string	Packet Name

Change Log

10 JAN 2003
Updated notes for TORA -Pa and -Po flags based on feedback by Daniel Brenner.
27 JAN 2003
Updated notes for new wireless trace events -Hs and -Hd flags based on questions from Qingjiang Tian.
28 JAN 2003
Added information about the old wireless trace format in older versions of NS2 and MIT Leach code format based on questions from Tahir Mahjabeen.
29 JAN 2003
Added -Pn to list of new wireless trace flags. tcp and and cbr use -Pn for packet type instead of -P.
20 FEB 2003
Added mobile node movement and energy trace formats.

If you have questions or comments, feel free to contact me at mailto:griswold.N0SPAM@acm.N0SPAM.org.