DARPA NMS Program: Network Models and Tools

Models

Categories

Contributor: people, institution, and primary point of contact (POC)

What is Modeled: network aspects addressed

Time Scale: time interval of application

Usage: what the model is used for

Network Scale: topology and traffic rate

Validation Data: type and network measured

Domain of Validity: where validation has shown model can be used

Software: language, platform, OS, availability, and how to get

Other: anything else

1 Fractional sum-difference (FSD)

Contributor

Bell Labs Math Research Center: Jin Cao, Bill Cleveland (POC), Yuan Gao, Don Sun.

What is Modeled

Packet arrivals and sizes on a link.

Time Scale

Models assume constant mean load, so applicable, as implemented, to time period over which change in diurnal variation of live traffic, so as a practical matter up to 5 minutes - 1 hour depending on the link.

Usage

Any open-loop engineering study.

Background (cross) packet traffic in simulation system.

Network Scale

Traffic on a link at any rate above 1 mbps.

Validation Data

Data: packet traces from links between

- San Jose and Seattle, MFN, OC48 PoS
- Mae West and NASA AMES, OC12 PoS
- Colorado State Ft. Collins and Internet access router, OC3 ATM
- New Zealand Internet Exchange router, 100 mbps Eth
- Bell Labs and Internet access router, 100 mbps Eth
- UNC Chapel Hill and Internet access router, 1 gbps Eth.

Domain of Validity

Traffic streams with 64 packets/sec and above

Small percentage of packets delayed on link input router (less than 15but model output can be put through simple queue to add delay.

Software

C code (available 3/15/03).

User specifies either

- all FSD model parameters
- traffic rate in packets/sec and FSD parameters of marginal distribution of packet sizes.

Other

Model could in principle accommodate changing mean load, perhaps as feedback from a control mechanism, but this is not part of current software.

2 FAST congestion control

Contributor

John Doyle (Caltech), Steven Low (Caltech), Fernando Paganini (UCLA)

What is Modeled

Equilibrium and stability of large networks under TCP congestion control and active queue management (AQM)

Time Scale

End-to-end congestion control time scale: 10 ms to 10 min

Usage

To analyze, design, optimize, and implement congestion control and AQM algorithms that are scalable, stable, robust, fair, with low loss and low delay, particularly for networks with high capacity and large (propagation) delay.

Network Scale

Arbitrary topology, size, capacity, routing, and load.

Validation Data

NS-2 simulations.

Live tests with Linux implementation on continental and intercontinental networks (CERN $\leftrightarrow$ Chicago $\leftrightarrow$ Sunnyvale $\leftrightarrow$ Baltimore $\leftrightarrow$ Caltech $\leftrightarrow$ Amsterdam, Netherlands) and briefly on Abilene, with link capacity from OC48 (2.5Gbps) to OC192 (10Gbps) and round trip delays from 80ms to 180ms.

Domain of Validity

Live tests were very successful and promising.

Arbitrary network, but particularly important for networks with high capacity (e.g. $>$ 1Gbps) and large delay (e.g. $>$ 100ms round trip time).

Software

Linux prototype of FAST TCP kernel (binary) available for testing early 2003.

Other

The Caltech FAST TCP Linux kernel has achieved 925Mbps (95% utilization) with a single TCP flow over an intercontinental path with round trip delay of 180ms, stably for more than an hour, using standard packet size. More info at netlab.caltech.edu/FAST/

3 Hybrid packet/fluid network with foreground/background traffic, using the HNS (hybrid Network Simulator) package

Contributor

K. Sohraby (UMKC), B. Melamed (Rutgers University), Yorai Wardi (Georgia Tech)

What is Modeled

The movement of workload in a telecommunications network, where both fluid and packet messages coexist in the simulation model.

Time Scale

Any simulation time horizon is admissible (in principle).

Usage

HNS models are fairly general, but the system is and particularly suited for speeding up the simulation of networks with foreground/background traffic. In such models, a small number of traffic flows (foreground traffic) is modeled in detail as packet flows subject to some protocol (e.g., TCP, ATM, etc.) The rest of the flows (background traffic) are viewed as competing traffic, which need not be modeled in detail, and is expected to consume a relatively small amount of simulation resources (CPU time and storage). To this end, background traffic is modeled as fluid, and its flows are aggregated to the extent possible (simulated fluid flows scale well under aggregation, unlike simulated packet flows). Finally, fluid flows are particularly suitable for simulating high

bandwidths, since their simulation complexity remains constant as the bandwidth increases.

Network Scale

Networks on the order of 1000 flows. Packet flows can be defined in any topology. Fluid flows are subject to feed-forward topology.

Validation Data

A number of synthetic network models with foreground/background traffic, consisting of hundreds of flows.

Domain of Validity

Link speeds of packet flows are 100 Mbps. Link speeds of fluid flows are 2 Gbps.

Software

HNS (Hybrid Network Simulator) package with a GUI, written in Java. User specifies a network model in a text file. The simulation is controlled from a GUI panel and produces dynamic statistics (text and graphics) for each simulation replication, as well as text summary statistics. Statistics are refreshed event-by-event or after each (prescribed) simulation interval. Developed on MS OS, but should be portable across platforms (as any Java code). To get a copy, contact Ben Melamed, melamed@rbs.rutgers.edu, 732-445-3128.

Other

Fluid flows scale very well in bandwidth.

4 An IntServ router with pricing

Contributor

Scott Jordan, UCI

What is Modeled

Aggregates of on/off fluid traffic sources that react to price, and an IntServ router that dynamically adjusts prices.

Time Scale

Flows are modeled as fluid flows, with the loss given by effective bandwidth formulae. Prices are adjusted dynamically over a medium time scale (slower than loss measurements but faster than significant changes in traffic load).

Usage

To obtain closed-loop control of traffic in an IntServ router.

Network Scale

A single router

Validation Data

None

Domain of Validity

None

Software

C

Other

None

5 A diffServ router with pricing

Contributor

Scott Jordan, UCI

What is Modeled

4 price sensitive traffic classes, a diffServ scheduler and packet drop algorithm, and a pricing adjustment algorithm.

Time Scale

Packets are transmitted as discrete events. Prices are adjusted dynamically over a medium time scale (slower than loss and delay measurements but faster than significant changes in traffic load).

Usage

To obtain closed-loop control of traffic in a diffServ router.

Network Scale

Single router

Validation Data

None

Domain of Validity

None

Software

Opnet + C

Other

None

6 Fluid flow models with an interface to packet-level simulator

Contributor

UCLA Rajive Bagrodia (POC), Mario Gerla, Medi Sanadidi, Mineo Takai, Junlan Zhou

UMass Amhurst Don Towsley

What is Modeled

Bahaviors of TCP with RED AQM

In addition to the fluid flow models originally developed by UMass, a new TCP protocol called TCP Westwood (proposed and developed as part of NMS) has also been modeled using the same fluid flow concept.

Time Scale

As short as time during which the traffic can be treated as fluid

Usage

Abstraction of background traffic in packet-level simulation

Network Scale

Any size; static routing is required at the moment

Validation Data

The model has been validated against packet-level simulation

Domain of Validity

Amount of traffic represented by the fluid flow model must be sufficiently large

Software

C code (translated from MATLAB)

Other

None

7 Discrete event (packet-level) models

Contributor

UCLA Rajive Bagrodia (POC), Mineo Takai, Junlan Zhou

What is Modeled

Complete protocol stack (physical layer to applications) Model library includes new TCP protocols developed under NMS

Time Scale

Packet level

Usage

The simulation environment includes various networking protocols to configure particular NMS scenarios

Network Scale

Up to thousands of wireless nodes, or hundreds of thousands of wired nodes for light traffic Up to a thousands of wireless nodes, or tens of thousands of wired nodes for heavy traffic

Validation Data

The simulation environment has been validated against real scenarios with 20-30 wireless nodes (mobile ad hoc networks)

Domain of Validity

Discrete event models are generically applicable

Software

C/C++ code Contact: rajive@cs.ucla.edu

Other

8 Fluid model for IEEE 802.11 DCF

Contributor

Department of Computer Science, University of Illinois at Urbana Champaign Hwangnam Kim and Jennifer Hou hkim27@uiuc.edu, jhou@cs.uiuc.edu

What is Modeled

Transmission activities under the distributed coordination function (DCF) of IEEE 802.11 in wireless LANs with and without the request-to-send/ clear-to-send (RTS/CTS) mechanism

Time Scale

Should be applicable to any time scale.

Usage

Fluid model based simulation to expedite the simulation for wireless LANs

Network Scale

Have been tested in the following scenarios: (i) a wireless LAN of up to 10,000 nodes; (ii) 20 wireless LANs each of which consists of 100 nodes; and (iii) 5 wireless LANs, each of which consists of 100 nodes and each node has 20 connections.

Validation Data

System throughput calculated from the fluid model has been compared against that obtained in packet-level simulation. System throughput obtained in fluid model-based simulation has also been compared against that obtained in packet-level simulation. The error falls within 2% of the system throughput value.

Domain of Validity

The model can be used in large-scale fluid-model based simulation for wireless LANs.

Software

Classes for performing fluid model based simulation has been implemented in ns-2 and JavaSim. Please contact Jennifer C. Hou (jhou@cs.uiuc.edu, +1 217 265 6329) for classes. Classes will also be included in the next JavaSim release.

Other

None

9 End-to-end measurement of cross traffic on the bottleneck link of an end-to-end path

Contributor

Department of Computer Science University of Illinois at Urbana Champaign Guanghui He and Jennifer Hou ghe@cs.uiuc.edu, jhou@cs.uiuc.edu

What is Modeled

Three methods for measuring, with the use of LRD characteristics of Internet traffic, cross traffic on the bottleneck link of an end-to-end path: prediction, reconstruction and interpolation. In the prediction-based method, we predict the future traffic based on the recent information obtained by probe packets. In the reconstruction method, we rebuild the entire cross traffic process with the information obtained by probe packets. In the interpolation method, we periodically send closely-spaced probe packet pairs to sample cross traffic, and infer cross traffic between two sampling points using interpolation.

Time Scale

The time interval of two back-to-back probe packet is set to be 5ms. This is determined by the probe packet size (L) and the bottleneck link bandwidth (B), i.e., L/B. In our studies, we use B=2Mbps and L=1000bytes.

In prediction and reconstruction methods, we send 21 probe packets in the interval of 100ms (5ms $\times$ 20) to get 20 samples, predict or reconstruct the cross traffic in the next 500ms (5 $\times$ 100ms). Then another 21 probe packets are sent to sample the cross traffic, and the process repeats. In the interpolation method, the period of closely-spaced probe packet pairs is set to be 500ms, the cross traffic in the interval is interpolated based on the two sampling values.

Usage

Can be used in end-to-end measurement of cross traffic. The objective is to infer cross traffic as accurately as possible, while not injecting a significant amount of probe packets into the network.

Network Scale

Simulation has been conducted in ns-2 on network topologies with a single bottleneck link (the dumb-bell topology) as well as with multiple bottleneck links. The network size has been varied from 100 to 2,000 nodes and the traffic has been generated with Internet traces.

Traffic rate won't affect the performance of the three methods, as what matters is the bottleneck link on the end-to-end path. First, the bottleneck link bandwidth may affect the interval between

two consecutive probe packets and hence influence the time scale at which we measure the cross traffic. Second, the dynamic change of the bottleneck link may lead to measurement errors in the methods.

Validation Data

The ns-2 simulation study indicates that (i) the prediction-based and reconstruction methods can give good mean measurement of cross traffic, while the interpolation method usually captures the instantaneous value of cross traffic better; and (ii) all three methods are adaptive to the dynamic change of cross traffic and are quite robust in the presence of multiple bottleneck links on an end-to-end path.

Domain of Validity

The model can be used in on-line network measurement.

Software

Measurement methods have been implemented in ns-2. Real-life daemons are being implemented in C. Please contact Jennifer C. Hou (jhou@cs.uiuc.edu, +1 217 265 6329) for details.

Other

All three methods are end-to-end based and does not require router support over the Internet.

10 Rapid Model Parameterization (RAMP)

Contributor

USC Information Sciences Institute (SAMAN project) Kun-chan Lan <kclan@isi.edu> John Heidemann <johnh@isi.edu> Tim Buchheim <buchheim@isi.edu> (POC)

What is Modeled

RAMP is a tool that support rapid parameterization of traffic models from live network measurements. By analyzing real traffic, RAMP estimate end-user behavior and network conditions to generate application-level simulation models. Currently we have developed structural source-level models for web and FTP traffic. The model parameters (user session inter-arrival time, object size, etc.) can then be used in traffic models to provide realistic simulation of real traffic.

Time Scale

With sufficient computing power (or for lower-speed traces), RAMP could parameterize the model in real-time. The idea is to incrementally update the output models (CDFs) as each flow arrives in a pipelined fashion, instead of computing all flows at once.

Usage

RAMP allows one to have more accurate models of web and FTP traffic than would be obtained from simple distributions. Unlike generic web models, it matches the model to observed traffic (page size distribution, etc) rather than a generic traffic distribution. Thus RAMP may capture site-specific variations that would not be captured in generic models based on wide-area traffic.

Network Scale

RAMP models arbitrary traffic rates over a single link.

Validation Data

See ”Rapid model parameterization from traffic measurement”
www.isi.edu/ kclan/paper.pdf

The data used in our study are from two sources. One was collected on the web server of Internet Traffic Archive. The other was recorded at a 100 Mbps Ethernet link connecting the Information Science Institute to the rest of the Internet.

Domain of Validity (where validation has shown model can be used)

To validate if RAMP accurately reproduce the traffic under study, we incorporated its output model into the ns-2 simulator and compared the result of simulation against the original traces. To understand if RAMP can perform as well as existing work in terms of generating realistic synthetic workload, we also compared RAMP with SURGE, a popular web traffic workload generator.

See ”Rapid model parameterization from traffic measurement”

www.isi.edu/ kclan/paper.pdf

Software (language, platform, OS, availability, and how to get)

Portable C code and Perl scripts (currently tested on Linux) uses pcap (tcpdump) or Coral Reef traces

Available as part of the current release of the ns network simulator (2.1b9a) as well as stand-alone. See
www.isi.edu/saman/ramp.html

Other

RAMP generates CDF files parameterizing the traffic. Examples which use ns to read these files and model the traffic are available..

11 The linear minimum mean square error based traffic predictor

Contributor

Department of Computer Science University of Illinois at Urbana Champaign Guanghui He, Yuan Gao and Jennifer Hou ghe@cs.uiuc.edu, yuangao@cs.uiuc.edu, jhou@cs.uiuc.edu

What is Modeled

Prediction of amount of future traffic at a router (or attainable throughput of a TCP connection at an end host) based on recent traffic measurements

Time Scale

For traffic prediction at an AQM router, the router measures and calculates the amount of incoming traffic every tau=20–50ms. The router keeps the history information for up to (20 x tau) seconds and makes prediction for the next tau interval. Our simulation and experimental studies show that tau should not be too small or too large, and the value of 20ms – 50ms was chosen. The number of history samples kept should be set to be 20 or more.

For traffic prediction at an end host, the end host keeps track of instantaneous attainable throughput according to received ACKs and calculates its attainable throughput every RTT interval (where RTT is measured by TCP). The end host keeps the history information for up to 20 x RTT time and makes prediction for the next RTT. Again, both our simulation and experimental studies show the number of samples kept should be set to be 20 or more.

Usage

We have demonstrated the use of traffic prediction in

(a) Modulating the magnitude of the packet dropping probability at an AQM router. We design a simple controller that takes the prediction result (i.e., the amount of traffic to arrive in the next few measurement intervals) as input to determine the packet dropping probability. The objective function used here is to stabilize the queue length at a desirable level.

(b) Inferring the optimal operational point at which a TCP connection should operate in the context of AIMD steady-state dynamics. We show that a TCP connection can reach the optimal point in one RTT without commencing MD phases (and hence without incurring packet losses), if all TCP connections are synchronized in making their congestion control decisions. This leads to significant performance improvement in terms of fast convergence to the optimal operational point, packet loss ratio, and attainable throughput.

Network Scale

ns-2 simulation has been conducted on networks with single-/multiple- bottleneck links and of size up to 20,00 nodes.

The network traffic rate does not affect traffic prediction.

Validation Data

The traffic predictor has been validated via ns-2 simulation with real Internet traffic traces. Usage of the traffic predictor to infer the optimal operational point of TCP has been validated with real-life implementation and experimentation for connections between Univ. of Wisconsin – Madison, U. California at Santa Barbara, U. Maryland at College Park, and Ohio State University.

Domain of Validity

The traffic predictor can be used to predict traffic parameters that exhibit LRD characteristics.

Software

Codes that are available are (i) C code for the traffic predictor; (ii) ns-2 simulation code for (a) and (b) in item ; and (iii) modified TCP code that implements (b) in item in FreeBSD and Linux. Please contact Jennifer C. Hou (jhou@cs.uiuc.edu, +1 217 265 6329) for code.

Other

None

12 MWM (Multifractal Wavelet Model)

Contributor

R. Riedi riedi@rice.edu (POC) R. Baraniuk richb@rice.edu V. Ribeiro vinay@rice.edu

What is Modeled (network aspects addressed)

Modeled: Aggregate traffic load on a link. Aspects addressed: By design, matches the correlation statistics on all dyadic time scales. Close match to the (possibly non-Gaussian) marginal distribution.

Time Scale (time interval of application)

MWM captures all time scales of interest, from microseconds to minutes or hours.

Usage (what the model is used for)

Rapid (linear computation time in the number of samples) traffic synthesis for simulation and verification.

Network Scale (topology and traffic rate)

MWM is versatile and can match traffic seen on networks with a large variety of topologies and rates. It models only the traffic on one link of a network.

Validation Data

LBL-TCP-3
ita.ee.lbl.gov/html/contrib/LBL-TCP-3.html

BC-pAug89
ita.ee.lbl.gov/html/contrib/BC.html

Auckland-II traces
pma.nlanr.net/Traces/long/auck2.html

Video traces from PhD thesis of Oliver Rose, University of Wuerzburg, Institute of Computer Science Series (1997), ”Traffic Modeling of Variable Bit Rate MPEG Video and its Impacts on ATM Networks”
www3.informatik.uni-wuerzburg.de/∼rose/files/MPEGtraces.zip

Domain of Validity

Used for fast traffic synthesis in laboratory network testbeds at Cetacean Networks, Inc., Portsmouth, (NH) Nortel (UK) Siemens (Germany)

and at various academic institutions for research purposes.

Software

Matlab code. C++ code.

spin.rice.edu/DARPA/soft.html

Other

None

13 Alpha-Beta (Traffic decomposition)

Contributor

R. Riedi riedi@rice.edu (POC) R. Baraniuk richb@rice.edu S. Sarvotham shri@rice.edu

What is Modeled

Modeled: Traffic on an access link is separated into a Gaussian component (beta component) which captures the bulk transfer, plus an erratic highly bursty component (alpha component) which is composed of only a few connections comprising a small part of the total load. Aspects addressed: The cause and origin of the disruptive bursts that are ubiquitous in today’s network traffic is traced back to simple topological properties of the network, more precisely the bandwidth and the round trip times of the active connections.

Time Scale

The alpha-beta decomposition captures all time scales of interest, from microseconds to minutes or hours.

Usage Predict traffic dynamics (burstiness, non-Gausianity) from network topology (round trip time distribution) and simple user behavior (transfer size distribution).

Network Scale (topology and traffic rate)

The alpha-beta is versatile and can explain traffic seen on large variety of topologies and rates. It is confirmed so far on access links of the internet.

Validation Data

LBL-TCP-3
ita.ee.lbl.gov/html/contrib/LBL-TCP-3.html

BC-pAug89
ita.ee.lbl.gov/html/contrib/BC.html

Auckland-II traces
pma.nlanr.net/Traces/long/auck2.html

Abeline IPLS-CLEV-20020814-090000-0
pma.nlanr.net/Traces/long/ipls1.html

Domain of Validity

To date primarily access links to the internet, as well as Abilene backbone. Additional testing on core links is planned.

Software

Matlab code C++

Software for the analysis is freely available. It does not come with a convenient user interface, however, and parameters have to be set by hand, currently. Assistance to interested parties is offered. Easy to use software is in development.

spin.rice.edu/DARPA/soft.html

Other

None

14 The "Veil" Emulation System.

Contributor

Georgia Tech. Kalyan Perumalla (POC), Richard Fujimoto.

What is Modeled

Veil is a novel approach to network emulation in which real distributed applications can be made to communicate over simulated networks. It consists of a system library for capturing, synchronizing and rerouting network data from unmodified distributed applications over a synthetic/virtual network. Among Veil's unique characteristics are: (a) it captures data from the application at the highest layer, even before they are converted to packets (b) it controls time advances to make the communication behavior highly repeatable (c) it doesn't require root/administrator access.

Time Scale

Time resolution is sufficiently fine to capture system overheads in microseconds for process context switches and network communication via BSD socket API. Network can be modeled at similar timescales at packet-level detail.

Usage

To perform very detailed network emulation. To test distributed applications over controlled network configurations. To inject high-fidelity application traffic into simulated networks to test network protocols.

Network Scale

Veil system can be retargeted to multiple different network simulators, so network scale dependent on user's choice of network simulator's realtime execution capability. Veil has been integrated with pdns (parallel & distributed ns). pdns has been tested to run thousands of network nodes (routers & end hosts) in realtime.

Validation Data

Tested with several distributed applications (both commercial as well as open source). Examples include OpenSSH, Netscape, thttpd, xtris, etc.

Domain of Validity

Many single-threaded distributed applications running on Linux or Solaris can be executed over Veil.

Software

Source code release (available 10/15/01).

www.cc.gatech.edu/computing/pads/nms/veil.html

ftp://ftp.cc.gatech.edu/pub/people/kalyan/veil/veil-current.tar.Z

Other

None

15 Discrete-Event Fluid Model of TCP

Contributor

Dartmouth College. David Nicol.

What is Modeled

The TCP protocol, using piece-wise constant rate functions, and routers that route both packet-oriented and fluid oriented flows. Fluid flows and packet flows are fully interactive.

Time Scale

The TCP fluid model admits to representation at time scales as small as a packet transmission time, and as large as a session time. The effective time scale is determined by factors such as RTT, link bandwidths, loss.

Usage

TCP-like background traffic to be used to support packet-level studies of other applications or protocols.

Network Scale

To be used on networks of the scale used when packet oriented studies are done.

Validation Data

Comparison of goodput, latency, and lost packet rate against pure packet simulation.

Domain of Validity

Wherever packet-oriented TCP is valid.

Software

Part of the DaSSFNet (C++) implementation, available from Dartmouth.

Other

None

16 Discrete event (packet-level) models

Contributor

RPI. Boleslaw Szymanski (POC), Yu Liu.

What is Modeled

Complete protocol stack (physical layer to applications) Model library includes extensions to NS, SSFNet for wired sub-networks and GloMoSim for wireless sub-networks to integrate the simulations running many parts of the network under any of the three simulators.

Time Scale

Packet level

Usage

The simulation environment includes various networking protocols to configure particular NMS scenarios.

Network Scale

Up to tens of thousands of wireless nodes, or millions of wired nodes for light traffic. Thousands of wireless nodes, or hundred of thousands of wired nodes for heavy traffic.

Validation Data

The simulation environment has been validated against scenarios with a few thousand wireless nodes (mobile ad hoc networks) and a few thousand of wired nodes under heavy TCP traffic.

Domain of Validity

Discrete event models are generically applicable.

Software

C/C++ and Java codes Contact: szymansk@cs.rpi.edu.

Other

None.

17 MPEG VBR Source Traffic

Contributor

John S. Baras, Majid Raissi-Dehkordi, Gun Akkor, University of Maryland

What is Modeled

A source model which generates the I, P and B

frames of the MPEG standard according to separate models for the sizes of each type of frame.

Time Scale

Can be set to any value.

Usage

Enables detailed simulation of MPEG traffic in NS-2.

Network Scale

NS-2's default capabilities. Traffic rate will depend on the settings of the parameters.

Validation Data

Original model model derived and explained in Marwan Krunz and Satish K. Tripathi, "On the characterization of VBR MPEG streams," Performance Evaluation Review (Proceedings of ACM SIGMETRICS'97 Conference), Vol. 25, No. 1, pp. 192-202, June 1997.

Domain of Validity

See above reference.

Software

CC file to be used as an NS-2 source model. File can be downloaded from

`www.isr.umd.edu/~majid/mpeg_ns2.`

Other

The model adds all the parameters necessary to describe the I,B,P traffic as variables in the C file.

18 Traffic Models for HTTP, FTP, TELNET and Email Applications.

Contributor

John S. Baras and Majid Raissi-Dehkordi, University of Maryland

What is Modeled

Simple models to capture the main characteristics of the above application/protocols written as Applications in the OPNET application definition node. Parameters are set according to the values extracted from previous traffic analysis projects.

Time Scale

On the order of msecs to seconds, but in general will depend on network parameters.

Usage

Enables more realistic simulation of the applications in OPNET.

Network Scale

OPNET's default capabilities. Traffic rate will depend on the settings of the parameters.

Validation Data

Data collected at the Hubrid Gateway of the DirecPC system (Hughes Network Systems)

Domain of Validity

Above network and similar networks.

Software

Files can be downloaded from (all.zip contains all files)

`www.isr.umd.edu/~majid/app_opnet.`

Other

None

19 Fluid Models for Congestion Control

Contributor

R. Srikant (UIUC, POC), S. Kunniyur, S. Shakkottai, S. Deb

What is Modeled

The dynamics of congestion control and active queue management mechanisms on the Internet

Time Scale

Applicable to long-lived file transfers, typically of the order of several minutes to a few hours.

Usage

Performance analysis of congestion control/AQM and to design new congestion control/AQM schemes

Network Scale

Independent of topology and traffic rate (fully scalable)

Validation Data

We are currently working with CAIDA to get data to validate the model. So far, extensive NS simulations have been used to verify the model.

Domain of Validity

The model is valid in regimes where the mean window size is at least 2 packets for each source.

Software

One of the results of the model, a new AQM scheme called AVQ, is now part of the NS code.

We are building a testbed to test the results from the model. This code will be made available on the web.

Other

None

Tools

Categories

Contributor: people, institution, and primary point of contact (POC)

Capability what the tool does

Usage: how the tool is used

Software: language, platform, OS, availability, and how to get

Other: anything else

20 SELFIS: Self Similarity Analysis Software Tool

Contributor

UC Riverside: Thoma Karagiannis, Michalis Faloutsos

Capability

The tool measures statistical properties of network behavior. In particular, the tool provides a comprehensive and systematic way to measure the long range dependence (LRD) characteristics.

Currently, it incorporates various long-range dependence estimation methodologies and the bucket shuffling methodology for long-range dependence detection.

Usage

The tool is easy to use, with a user-friendly interface.

Software

The tools is developed in Java for portability, and is a stand alone tool: it does not require any additional commercial software.

Other

Future releases will include time-series processing algorithms and transforms, such as wavelets, Fourier transform, stationarity tests and smoothing algorithms. In addition, the next release will include FGN and FARIMA processes generators to simulate long-range dependence in time-series.

21 Maya: A Multi-Paradigm Network Modeling Framework

Contributor

UCLA: Rajive Bagrodia, Mineo Takai, Junlan Zhou and Zhengrong Ji

Capability

This multi-resolution, multi-paradigm capability for network analysis is realized by a combination of novel techniques including parallel discrete event simulation models, fluid flow abstractions and other aggregation techniques. The objective is to permit performance prediction of networks with millions of nodes with bounded inaccuracy, while providing the capability for online analysis

of smaller networks.

Usage

The framework consists of multiple NMS tools including QualNet, RTI and Backplane by Georgia Tech and a fluid flow model by UMass Amhurst. It will be available to NMS program participants in a form of patch to base NMS tools.

Software

`pcl.cs.ucla.edu/projects/maya`

Other

None.

22 Workload Generator for Network Simulations

Contributor

Rensselaer Polytechnic Institute: Murat Yuksel, Biplab Sikdar, Kenneth S. Vastola, Bolek Szymanski

Capability

The tool provides a platform to generate realistic traffic for ns based simulations. We provide two self-similar traffic generators one based on the superposition of Markov Modulated Poisson Processes and the other based on fractal point processes. In addition, the tools allows for the generation of traffic with application specific characteristics (telnet, web, ftp and smtp),

Usage

The tools is developed for the network simulator ns. It needs to be integrated into ns through some very simple steps which are outlined in a file included with the distribution. This tools is already being used by a number of research groups.

Software

`networks.ecse.rpi.edu.edu/~bsikdar/software/all.tar.gz`

Other

None.

23 JavaSim

Contributor

Jennifer C. Hou Department of Computer Science University of Illinois at Urbana Champaign jhou@cs.uiuc.edu +1 217 265 6329

Capability

JavaSim a component-based, compositional network simulation, emulation, and protocol synthesis environment, called JavaSim, that is truly extensible, reusable, and allows debugging/diagnosis at the component level, rather than at the implementation level. There are several salient features of JavaSim:

(F1) Loosely-coupled, component-based programming model. We propose a component-based software architecture, called the autonomous component architecture.

The ability to handle data in independent execution contexts, along with the fact that components are loosely coupled and only bound to one another at system integration/implementation time, enables a component to be autonomous (and hence the name of the architecture). A component can be designed, tested, and reused in other software systems with the same contract context, in much the same way an IC chip does in the IC hardware design. This is the key reason why JavaSim avoids the well-known hyperspaghetti subsystems phenomenon (which refers to the situation in which modules in a system are tightly coupled such that it is almost impossible to extract some of the codes for reuse, and JavaSim classes are truly extensible and reusable.

(F2) Implementation of an (almost) complete suite of Internet Integrated/Differentiated/Best-effort Services protocols and classes for the FCS demo: We have laid out a a generalized packet-switched network model on top of the proposed architecture. The generic network components and their contracts are defined and implemented that capture their fundamental features and yet are flexible enough to accommodate future technology advances. We realize the model in Java (and hence the name JavaSim), and include an (almost)

complete suite of Internet protocols and classes needed for the FCS demo in the current implementation. With all the classes available, composition of protocol stacks and simulation under different network scenarios can be realized in a plug-and-play fashion.

(F3) A rich, Tcl and Java/Swing-based user interface to facilitate auto-configuration and on-line monitoring: We have implemented in JavaSim a script-based user interface and a graphic user interface (that leverages Java AWT and Swing) to facilitate construction of network scenarios, composition of protocol stacks, and result display at run time. In particular, we integrate into JavaSim the pure Java implementation of Tcl 8.0 with Tcl/Java (called Jacl) and Python. This enables access to Java objects, such as creating an object from a certain Java class, and invoking a method or accessing a field variable of a Java object, in the Tcl environment.

(F4) We have realized (a form of) network emulation by developing a Java-compliant socket layer, on top of which real applications (e.g., WWW traffic and audio/video streams) can be ported. The socket layer will give applications the illusion that they are interfacing with the operating system, rather than with a virtual network environment.

(F5) To allow extremely large scale network simulation in JavaSim, we have enabled JavaSim to conduct parallel and distributed simulations by use of Java RMI on a Myrinet based testbed.

(F6) We have studied in the framework of JavaSim the issue of applying fluid model-based simulation to IEEE 802.11-operated wireless local area networks. We have derived a fluid mode that captures the data transmission activities between two successful frame transmissions. Based on the above fluid model, we have implemented in JavaSim fluid model-based simulation with the use of the time stepped simulation technique.

(F7) We have also implemented mixed-mode simulation classes in JavaSim to study the throughput behavior of TCP congestion control (that forms a feedback control system together with the queue management strategy used in routers). The foreground traffic (whose dynamic is being studied) is simulated at the packet-level and the background traffic (that comprise of possibly hundreds of flows) is modeled with a fluid model. We adopt the time-stepped simulation technique and capture the background traffic as "fluid chunk packets" that are relayed periodically, and capture the effect of the foreground traffic in the fluid model that characterizes the background traffic (and vice versa).

Usage

JavaSim can be used as an evaluation and validation environment for network protocol design and synthesis. It can be easily extended to new network architectures and applications. (We are currently in the stage of implementing grid networking classes in JavaSim.) With the parallel engines and fluid/network calculus models available, JavaSim can also be used to carry out large-scale network simulation. Finally with the Java-compliant socket layer available, JavaSim can also be used for network emulation.

Software

`www.javasim.org`

Other

None.

24 NS Fluid Network Module

Contributor

Yu Gu, Yong Liu, Francesco Lo Presti, Vishal Misra, Don Towsley

Capability

The fluid network module can be used in ns simulations to simulate network segments which consist of a large number of links and high volume of traffic. It is based on analytical dynamic models of network components, including TCP sources at the edges and Active Queue Management schemes in the core. By solving fluid models efficiently, the fluid network modules can provide accurate information about the key performance metrics of large network segments, such as packet delay, packet loss, flow throughput, with low com-

putation cost. It has been used to simulate networks with thousands of links and millions of TCP flows in real time.

Usage

Fluid network module provides a way to conduct ”hybrid” simulations of large IP networks: some parts of the network, which are normally small and for which packet level details are desirable, are simulated using packet level discrete event simulation functions provided by ns, some parts of the network, which are normally large and no packet level detail is necessary, are simulated by fluid network module. Packet level simulation interacts with fluid simulation through traffic/message exchanges and event scheduling: at entry points of fluid network segments, packets generated by the packet level simulation are converted into fluid traffic and fed into fluid models. Fluid modules are responsible for solving fluid models to provide end-to-end packet loss and delay information for all packets traversing fluid network segments. This information is used to transfer packets from fluid network segments to the packet-level simulation. Time is advanced within the packet-level and fluid simulations fluid simulations in a synchronized manner so that packets will always be scheduled out from fluid network segments in time and in order. Hybrid simulation greatly reduces computation cost of pure packet level ns simulation.

Software

We use standard ns programming interface for the fluid module. It is straightforward for users to set up a fluid network topology and traffic by configuring fluid network objects, such as fluid net, fluid link, access points, etc. Our ns fluid network module and documentation is publicly available for study/research.

Other

None.

25 Wireless QoS Routing and Scheduling Protocol (WQRSP)

Contributor

Tools for Smart Networks Group at University of California Berkeley: Venkat Anantharam, Eric Chi, Antonis Dimakis, Linhai He, Bill Hodge, David Jaffe, Zhanfeng Jia, Eduardo Magnana, Daniel Morato, Jun Shu, Wilson So, David Tse, Teresa Tung, Pravi Varaiya, Jean Walrand

Capability

The system solves the problem of routing flows with QoS requirements (minimum bandwidth) in shared medium wireless networks. For the routing, the global network is divided into different routing domains with independent wireless channels. The domains are interconnected through gateways, special nodes in each domain with links to other domains. These gateways are responsible for optimizing the routing for the QoS flows inside their domain. They run a server software that computes the optimal solution using an Integer Linear Program and configures the result in the local domain nodes. The computation/configuration for other domains in the flow path is distributed (for scalability reasons) to gateways in those domains. The formulation of the linear program takes into account the sharing of the wireless medium among all the flows using each domain. Inside each domain a distributed protocol implemented in every wireless ad-hoc node coordinates the amount of traffic that each node sends in order to avoid the saturation of the channel. This way best-effort traffic can use the bandwidth left by the traffic with QoS requirements without congesting the channel and so reducing the quality for QoS flows.

Usage

Our program can be used to route application flows with bandwidth requirements in an ad-hoc wireless network. We assume that the network has a natural hierarchical structure which allows the nodes to be grouped into clusters.

Software

A C program implements the above mentioned routing algorithm as a stand alone program. The program takes several static files as inputs. One file describes the nodes and the interfaces of each

node. Another file describes the wireless links between any pair of nodes together with the Signal-to-Noise Ration (SNR). The output of the routing algorithm is a list of routes to be installed at each of the routers.

Other

URL:

`www.cs.berkeley.edu/~so/NMS/WQRSP/`

Paper (in submission):

`www.cs.berkeley.edu/~so/NMS/WQRSP/wqrsp.ps`

26 HNS (Hybrid Network Simulator)

Contributor

K. Sohraby (UMKC), B. Melamed (Rutgers University), Yorai Wardi (Georgia Tech)

Capability

User specifies a network model in a text file. The simulation is controlled from a GUI panel and produces dynamic statistics (text and graphics) for each simulation replication, as well as text summary statistics. Statistics are refreshed event-by-event or after each (prescribed) simulation interval. Developed on MS OS, but should be portable across platforms (as any Java code).

Usage

HNS models are fairly general, but the system is and particularly suited for speeding up the simulation of networks with foreground/background traffic. In such models, a small number of traffic flows (foreground traffic) is modeled in detail as packet flows subject to some protocol (e.g., TCP, ATM, etc.) The rest of the flows (background traffic) are viewed as competing traffic, which need not be modeled in detail, and is expected to consume a relatively small amount of simulation resources (CPU time and storage). To this end, background traffic is modeled as fluid, and its flows are aggregated to the extent possible (simulated fluid flows scale well under aggregation, unlike simulated packet flows). Finally, fluid flows are particularly suitable for simulating high bandwidths, since their simulation complexity remains constant as the bandwidth increases.

Software

HNS (Hybrid Network Simulator) package with a GUI, written in Java.

Other

Fluid flows scale very well in bandwidth.

27 OSC APIs

Contributor

SAIC: Ron Nolte, Luke Burns

Capability

Enables network simulation to integrate into real-time DoD operations via two aspects:

a. Enables input of sniffed or other network traffic specifications while simulation is running, and at several levels of detail. b. Enables output from simulation of metrics for QoS analysis leading to user satisfaction as a final output metric.

Usage

By developers – Insert calls to the API, per documentation, into network simulation engine. Most appropriate for discrete event simulation. May be adapted to more aggregate simulation. Input and output portions of the API may be used separately.

Software

C code.

Other

None.

28 OSC Framework

Contributor

SAIC: Ron Nolte, Ken Funk, Brian Merrell

Capability

Dynamically create and control task flows that use network modeling and simulation for a variety of purposes. Key uses include: network planning,

real-time network tuning, and real-time network QoS performance analysis and display. Can display user satisfaction levels and underlying technical metrics. Can distribute the task onto many computers.

Usage

By operators – Start the framework for a prespecified task flow and use the framework, via its GUI for that task flow. Start multiple instances of the framework for different task flows. Move the results from to flow by point each flow that the electronic results of the previous flow. Define new task flows. E-mail new task flow definitions to other operators so that they use them.

Task flow definition includes selection of machines to run various sub-tasks on. It includes selection of various tools for each sub-task. For example, for simulation can select among OPNET, QUALNET, etc. Define which data should be displayed and how frequently it should be updated.

By developers – Use the modular framework and interface definitions to insert new tools. Whether a tool is a simulator, and traffic data source, or display, there are documents that define the interfaces required and their forms. A new tool can be modified or wrapped into the framework. Once it meets specifications, the framework can be told of the new tool via the administration menu of the framework. Operators can then see the new tool and insert into task flows.

May be used whole or in parts.

Software

Integrated tcl/tk/blt. Also includes c code, and some GOTS. Runs on unix and Linux. Most of it also runs on Windows. Tools that are integrated can be in any language and can be COTS, GOTS or custom.

Other

None.

29 USF: Unified Search Framework and RRS: Recursive Random Search

Contributor

Rensselaer Polytechnic Institute: Tao Ye, Shivkumar Kalynaraman

Capability

The tool provides a general optimization platform for engineering black-box optimization problems, e.g., network protocol parameter optimization. The tool includes various search methods, such as, random sampling and pattern search, as the building blocks. A tailored optimization algorithm can be built by simply combining a selection of building blocks according to the features of the underlying problem. The search methods are run in parallel in USF and each is allocated with a certain amount of computing resources among these methods. To make the best of available computing resources, e.g., a network of work stations, the tool includes the mechanism to distribute the experiments to these computing resources and run these experiments in parallel.

The Recursive Random Search(RRS) is an optimization algorithm especially designed for engineering optimization problems such as network protocol parameter optimization. This algorithm is efficient, scalable to high dimensions and robust to the effect of noise in objective functions. It is also able to automatically exclude negligible parameters from the optimization process. RRS has been integrated into USF and can be used as one component in the tailored hybrid optimization algorithm.

Usage

The USF is a stand alone optimization tool developed in Java. It can distribute experiments, e.g., network simulation, to a network of workstations using use Java RMI technology. To use this feature (optional), Java registry service has to be run on each workstation. It does not require any additional commercial software. The tool is easy to use, with a user-friendly interface.

Software

The software is available for download at this webpage:

`networks.ecse.rpi.edu.edu/~yet3/research/usf.html`

Other

None.

30 Federated Simulations Development Kit (FDK)

Contributor

Georgia Tech

Capability

High Performance Run Time Infrastructure software customized for efficient execution on parallel/distributed computing platforms such as clusters of workstations and shared memory multiprocessors.

Usage

This software provides services such as communication and synchronization for federated distributed simulations. Other software (e.g., network simulation.emulation backplane software executes over the FDK RTI. High Level Architecture (HLA) interfaces are included in the package as well as Jane, a client/server package enabling remote execution of distributed simulations.

Software

Source code in C, C++, and Java is provided.

Other

The original FDK software pre-dates the NMS program, however NMS funds have been used to enhance and optimize the FDK package, e.g., to realize an efficient implementation executing over networked SMPs.

31 The Dynamic Simulation Backplane

Contributor

George F. Riley, Assistant Professor, Georgia Tech; Richard M. Fujimoto, Professor, Georgia Tech; Kalyan Perumalla, Research Scientist, Georgia Tech; Donghua Xu, Graduate Research Assistant, Georgia Tech.

Capability

The Dynamic Simulation Backplane is a set of run–time interfaces that allow dissimilar network simulation environments (such as ns2 and GloMoSim) to negotiate a common set of capabilities, and to exchange meaningful information during the course of the simulation. Thus, models created on one environment and be combined with models created in a different environment, resulting in an integrated simulation utilizing the features of each simulator as needed.

Usage

The Dynamic Simulation Backplane software consists of a set of C++ subroutines which must be called by the existing network simulators. A set of initialization routines allow the simulators to specify their capabilities (such as ”support of IPV4 checksum field”), and to specify the required/optional capabilities of peer simulators. The Dynamic Simulation Backplane negotiates a common subset of capabilities, and informs all simulators of the results.

Later, each simulator uses the Export features of the backplane to create a free–field representation of packet level data to be exchanged with other simulators. The relevant information is stored in a Type/Length/Value array, which contains only relevant information for that packet. When receiving a set of information from a peer, the simulator uses the Import features of the backplane to reconstruct a simulated packet in a format meaningful to that simulator.

Software

The Dynamic Simulation Backplane is coded completely in C++ and is publicly available in open–source form.

Other

The Dynamic Simulation Backplane has been successfully integrated into both ns2 and GloMoSim, and demonstrated using a federated simulation of

a wireless in GloMoSim, and a supporting wired infrastructure in ns2.

32 The Georgia Tech Network Simulator (GTNetS)

Contributor

George F. Riley, Assistant Professor, Georgia Tech, and students.

Capability

The research effort to design and implement the Georgia Tech Network Simulator was started in the Summer of 2002. The purpose of the effort was to create a new network simulation environment that could be used in cases where existing network simulation environments lacked the capabilities to achieve the simulation goals desired. GTNetS has a number of basic design goals, as follows.

a) The simulator is written entirely in C++. To use the simulator, the simulationist creates a C++ main program, instantiating C++ objects to represent the various network elements comprising the simulation. Most of the supplied C++ objects that encapsulate the functionality of network elements use virtual functions, to allow easy extension and modification of behavior. For example, there is a single virtual base class describing the behavior of a Queue. All queuing methods, such as DropTail and RED, use a subclass of queue to define their behavior.

b) The simulator is designed like real networks are designed. In GTNetS, there is a clear distinction between nodes, interfaces, links, and protocols. Packets consist of a list of Protocol Data Units (PDUs), which is created and extended while a packet moves down the protocol stack. When moving up the stack, each protocol layer removes and processes the corresponding protocol header in a fashion closely modeling a real protocol stack. Interfaces have associated IP Addresses and IP Address Masks. Nodes have one or more IP Addresses based on the IP Addresses assigned to the interfaces. Layer 4 endpoints are bound to port numbers, either well know fixed values or transient ports, just like real layer 4 endpoints. Connections between layer 4 endpoints are by IP Address and Port Number, in a fashion nearly identical to actual protocols. In general, when faced with a design decision for the simulator, a design similar to actual networks was chosen whenever possible.

c) The simulator is designed from the beginning to be scalable and distributed. One major design feature is the use of Ghost Nodes that act as placeholders in a distributed simulation. By using Ghost Nodes , each simulator has a complete picture of the overall topology of the simulated network, but only has full state representations for those nodes mapped to that simulator in the distributed simulation.

d) The simulator has fine grained control over logging of simulation events. When running large–scale simulations, the total number of simulation events processed can be excessively large, potentially trillions or more. Given this large number of events, it is not practical to log a complete record (on disk files) of all simulation events for post–analysis. With GTNetS, the events that are logged are completely controlled by the simulation script, with fine grain control. For example, the simulation program can request that log entries be created only for layer 4 protocols at specific nodes in the topology. Further, all logging can optionally be disabled at selected nodes (such as interior routing nodes).

e) A protocol graph is used to map protocol numbers to protocol objects, in a fashion similar to actual protocol stacks. At initialization time, the IPv4 implementation registers the use of protocol number 0x800 at layer 3. When a packet is later received at any layer 2 protocol, the layer 3 protocol number is requested from the protocol graph, and a pointer to the IPv4 instance is returned (for protocol number 0x800). Layer 4 works similarly, with TCP registering protocol number 6 and UDP registering number 17.

f) The simulator uses MAC addresses for all link endpoints, and supports both Layer 2 broadcasts and Layer 3 broadcasts.

g) The TCP model uses a client/server paradigm

identical to real TCP implementations. To create a web server for example, a single TCP endpoint is assigned to a node, bound to port 80. Connection requests (SYN packets) received by this endpoint cause the creation of a new TCP endpoint which responds to the connection request, and the original TCP on port 80 continues to listen for SYN packets.

h) Applications use an interface very similar to the familiar sockets interface to create and manage layer 4 connections. There are equivalent functions to the familiar connect , listen , send , sendto , and close just to name a few. The major difference is the use of upcalls for received data, rather than the more familiar blocking read calls from the socket library.

i) GTNetS has full support for modeling data contents as well as data length when moving data between layer 4 protocols. Most current simulators, including ns2, do not provide an easy way for applications to specify and receive data contents. Modeling data contents is essential for a number of networking research simulation experiments, such as studying the behavior of large–scale peer–to–peer networks.

j) GTNetS uses nv routing as the default packet routing mechanism. While this is not in fact the way existing networks are designed, the resulting savings in simulator memory is believed to be a beneficial tradeoff.

k) Simulation models for a number of different random number generators are provided, including exponential , pareto , normal , uniform , empirical , and constant .

The current capabilities of GTNetS include: a) Application models for web browsers, web servers, TCP client/server applications, On–Off data sources, FTP clients/servers, and constant bit rate sources. b) TCP models for Tahoe, Reno, NewReno, and SACK. c) Queuing disciplines DropTail and RED. d) UDP models e) nv and static routing methods f) Layer 2 protocols for 802.3 (Ethernet) and 802.11 (wireless) g) Layer 3 model for IPv4 h) Link objects for point–to–point links, CSMA-CD LANs, and 802.11 wireless links. i) Node mobility models for wireless including random waypoint and specified waypoint. j) A graphical display of node mobility using X-Windows. k) A number of stock objects for creating well–known topologies, such as star , dumbbell , grid , and tree .

Usage

The GTNetS simulator is a C++ programming environment. The simulation topology and data flow definitions are described using C++ objects, in a fashion similar to the use of TCL programs to construct ns2 simulations. The simulator provides pre–defined behavior for a large number of stock objects as described, which facilitate the creation of large models with minimal effort by the simulationist.

Software

The GTNetS simulator is released as open–source, with complete source code. Any standards–conformant C++ compiler can compile and link GTNetS. It has been tested with GNU C++ 2.96, GNU C++ 3.2, Sparc Solaris CC, and HP CC.

Other

The simulator is not presently (as of April 1, 2003) released to the networking research community, primarily due to the lack of complete user–level documentation. We plan to release the product as open–source to the community at large by the end of the Spring semester, 2003.

33 Dosdbell

Contributor

USC Information Sciences Institute (SAMAN Project) Alefiya Hussain (hussain@isi.edu) John Heidemann (johnh@isi.edu) Christos Papadopoulos (christos@isi.edu) Tim Buchheim (buchheim@isi.edu) (POC)

Capability

dosdbell creates a compact dumdbell topology from a larger network topology. The attackers, victim and the active network hosts are represented using a balanced dumbbell with a single bottleneck link. This greatly simplifies the route computation process and facilitates running simu-

lations with large number of nodes.

Usage

dosdbell abstracts the topological details present in the network and creates a balanced dumbbell. The number of hops between any arbitrary pair of host in the network is specified as probability density function (pdf) within dosdbell and parameters for the dumbbell topology, cross-traffic and attack traffic are inputed from a file.

Software

dosdbell is integrated into the current release of ns-2 network simulator, available at

`www.isi.edu/nsnam/ns/.`

Other

Web page:

`www.isi.edu/saman/dosdbell.html`

34 Dosreduce

Contributor

USC Information Sciences Institute (SAMAN Project) Alefiya Hussain (hussain@isi.edu) John Heidemann (johnh@isi.edu) Christos Papadopoulos (christos@isi.edu) Tim Buchheim (buchheim@isi.edu) (POC)

Capability

dosreduce creates a reduced network topology from a larger inet topolgy based on intersection of attacker and crosstraffic paths. The nodes and links that do not intersect with the attack tree are pruned to create a reduced network topology.

Usage

The inet topology with the number of attackers and active connections is inputed to dosreduce. The tool uses BFS to find the attack tree and the cross traffic paths. The output is a set of nodes and links that belong to this intersection.

Software

dosreduce is available in the current release of ns-2 network simulator, available at

`www.isi.edu/nsnam/ns/`

Other

Web page:

`www.isi.edu/saman/dosreduce.html`

35 Asim

Contributor

USC Information Sciences Institute (SAMAN Project) Debojyoti Dutta ¡ddutta@isi.edu¿ John Heidemann ¡johnh@isi.edu¿ Tim Buchheim ¡buchheim@isi.edu¿ (POC)

Capability

Asim performs a very fast analysis of network scenarios to provide approximate answers much faster than detailed packet-level simulations. Although the fast analytical approach results in some error (typically 10%) it is 10-300 times faster than a detailed simulation.

Usage

sim is particularly useful as a pre-filtering tool for selecting scenarios. For example, if one is looking for a particular set of parameters to optimize performance on a network, one might want to run simulations with many slight changes in parameters in order to find the best settings. With detailed simulation, this might take far too long. Asim allows one to run a large number of very fast simulations to provide approximate results. Those simulations which meet the target criteria could then be put through a detailed packet-level simulation if more accurate results are desired.

Software

Asim is integrated into the current release the ns-2 network simulator, available at

`www.isi.edu/nsnam/ns/`

Other

Web page:

`www.isi.edu/saman/asim.html`

Debojyoti Dutta, Ashish Goel, and John Heidemann. Faster Network Design with Scenario Prefiltering. In Proceedings of the International Symposium on Modeling, Analysis and Simulation of

Computer and Telecommunication Systems, pp. 237-246. Fort Worth, Texas, USA, IEEE. October, 2002.

`www.isi.edu/~johnh/PAPERS/Dutta02d.html`

36 Genesis-SSFNet

Contributor

RPI: Bolek Szymanski, Yu Liu

Capability

Enables packet-level, fast, computation and memory distributed network simulation under SSFNet using Genesis fixed-point-solution approach. UDP, TCP and BGP protocols are supported.

Running the simulation on n processors often yields n-times speedup with the precision of the results in the order of 5-10% in the traffic intensity.

Usage

By simulation designers – define network parts in the network configuration description and assign the simulation to the n distributed computers.

Software

Java code.

Other

None.

37 Genesis-NS2

Contributor

RPI: Bolek Szymanski, Yu Liu, Anand Sastry

Capability

Enables packet-level, fast, computation distributed network simulation under NS2 using Genesis fixed-point-solution approach. UDP and TCP protocols are supported. Running simulation on n processors often yields n-times or higher speedup with the precision of the results in the order of 5-10% in the traffic intensity.

Usage

By simulation designers – define network parts in the network configuration description and assign the simulation to the n distributed computers.

Software

C++ code.

Other

None.

38 Genesis-GloMoSim

Contributor

RPI: Bolek Szymanski, Kiran Mandani

Capability

Enables packet-level, fast, computation distributed network simulation under SSFNet using Genesis fixed-point-solution approach. Mobility and MAC protocols are supported as is interoperability with wired domains. Running the simulation on n processor often yields n-times speedup with the precision of the results in the order of 5-10% in the traffic intensity.

Usage

By simulation designers – define network parts in the network configuration description and assign the simulation to the n distributed computers.

Software

C++ code.

Other

None.

39 CoralReef: A Passive Network Monitoring and Analysis Suite

Contributor

CAIDA

Capability

CoralReef is a suite of device drivers, libraries, and applications that enable UNIX-like systems

to collect and analyze network traffic. CoralReef supports a rich set of features and includes APIs and hooks at every network layer. CoralReef makes it easy to develop new monitoring applications with minimum effort.

Usage

The bundled "crl" family of applications processes raw traffic data. These applications can capture network traffic to ".crl" or tcpdump trace files; report hardware and link configuration details; report timestamps and inter-arrival time information for packets or ATM cells; print multiple layers of protocol headers and payloads of packets and ATM cells; report packet and byte rates, packet length histograms; and more.

Software

The libcoral library provides a uniform interface to "crl" applications for reading traffic data from live ATM and POS devices, live network interfaces, or trace files (including tcpdump files) while hiding irrelevant details from the user.

Other

CoralReef is available for download at:

`www.caida.org/tools/measurement/coralreef`

For more information, please send email to coral-info@caida.org.

40 NeTraMet: A Toolkit for Measuring Network Traffic Flows

Contributor

Nevil Brownlee, CAIDA

Capability

NeTraMet is a flexible, efficient tool for analyzing network traffic flows.

Usage

The NeTraMet toolkit consists of meters that gather data about packets, consolidating this to provide flow data; meter readers that retrieve flow data from meters using SNMP; managers that coordinate the activities of meters and meter readers; Simple Ruleset Language (SRL) compiler that helps produce meter configuration files; and flow data utilities that compute attribute-value differences between successive readings in a flow data file.

Software

NeTraMet is a GPL open-source implementation of the Real-time Traffic Flow Measurement (RTFM) architecture, as defined by the RTFM Working Group of the Internet Engineering Task Force (IETF). NeTraMet's SRL language makes it easy for users to configure meters to collect specific end-point attributes from flows, enabling a powerful and efficient way to gather the data for a wide range of analyses, including accounting, workload characterization, and web session performance. NeTraMet's NeMaC, a combined manager/Meter Reader, downloads compiled SRL rulesets to meters, collects data from meters at regular intervals, and writes data to a file.

Other

NeTraMet can be downloaded from

`ftp.caida.org/pub/iawg/netramet/.`

Examples and more detail can be found at

`www.caida.org/tools/measurement/netramet/`

and

`www.caida.org/analysis/workload/netramet/.`

41 Dnsstat

Contributor

CAIDA

Capability

The dnsstat application watches for DNS queries on UDP port 53. It counts numbers of messages and numbers of queries, aggregated by any of source IP, destination IP, opcode, query type, query class. The subjects of queries are never recorded.

Usage

To collect accurate statistics on a specific nameserver (or client), it must be run on an interface that sees all DNS messages to that server (or from that client).

Software

dnsstat is one of the CoralReef family of applications.

Other

dnsstat can be downloaded from:

`www.caida.org/tools/utilities/dnsstat/.`

42 Iffinder

Contributor

CAIDA

Capability

iffinder discovers which IP interfaces belong to the same router.

Usage

iffinder works by sending a ”probe” UDP packet to an unused port on an interface address. Many routers will reply to such a packet with an ICMP PORT UNREACHABLE error with the source address set to that of the interface on the unicast route back to the prober. So probing one interface and getting this error from a different interface proves that the two interfaces belong to the same network node.

Software

iffinder sends a series of 1 to 3 probes to a particular interface. To avoid conflicts with traceroute processes and other iffinder processes running on the same host, iffinder chooses its source port in the same way as traceroute, as a function of its process id. Iffinder does not probe unroutable addresses. Whenever iffinder encounters a new interface, it adds that interface to its list of probe targets.

Other

iffinder is available by sending an email request to kc@caida.org.

43 NetLets: Network Daemons for Low End-to-End Message Delays

Contributors

Oak Ridge National Laboratory: Negewara S. V. Rao

Capability

This tool consists of application-level daemons that implement an overlay network of in-situ instruments on wide-area networks to collect delay measurements, compute quickest paths and route messages. The daemons realize single and multiple paths among themselves without explicit support from the underlying network routers.

Usage

Netlet daemons are application-level programs that are executed from the command line.

Software

NetLet daemons are developed in C++ using sockets under linux/unix environments. These daemons form a standalone tool and do not require additional commercial software.

Other

Netlets are currently in the final stage of testing over the Internet and will be released to public within next few months.

44 Stable Transport Daemon

Contributors

This tool provides a stable transport flow from a source to a destination on the Internet at a specified level, typically much below the peak bandwidth. This flow can be used to implement control channels for applications such as computational steering and remote instrument control.

Capability

This daemon is an application-level program that can be executed from the command line or called by another application.

Usage

This daemon is an application-level program that can be executed from the command line or called by another application.

Software

This tool is developed in C++ using sockets under linux/unix environments. This is a standalone tool and does not require additional commercial software.

Other

We expect to release this tool to the public within next few months.

45 SSFNet

Contributors

Renesys Corporation (James Cowie, Andy T. Ogielski, BJ Premore), Dartmouth College (David Nicol, Michael Liljenstam), and many other collaborators.

Capability

Open-source Java models of protocols (IP, ICMP, TCP, UDP, BGP4, OSPFv2, HTTP and others), network elements (hosts, routers, links, LANs), traffic generating applications and assorted support classes for realistic multi-protocol, multi-domain Internet modeling and simulation.

Usage

Modeling, simulation and analysis of large internetworks consisting of many autonomous systems, such as multiple Internet service provider backbones and their customers' networks. SSFNet provides a unique scalable infrastructure for collection of streaming network data from many locations, and simulation and analysis of attacks on the routing infrastructure, including worm propagation models. Enables simulation and analysis of dynamic interactions of routing protocols (BGP for inter-domain routing, OSPF for intra-domain routing) under various failure scenarios.

Software

Downloads and extensive documentation are available from www.ssfnet.org

Other None.

46 Renesys GRADUS Services: Real Time Monitoring of Global Internet Routing

Contributors

Renesys Corporation: Andy Ogielski

Capability

GRADUS is an internetwork monitoring service, delivered over the World-Wide Web. GRADUS subscribers monitor evolving routes to their networks in real time as seen from multiple vantage points in the Internet. They set real-time alarms on selected routing failures and other conditions, and perform interactive routing analyses, drill-down and forensics using an extensive repository of archived BGP update traffic.

Usage

Global BGP monitoring. Typical users include:

1. Network operators concerned primarily with network performance and infrastructure management.

2. Analysts focused on security, auditing and forensic analysis of network failures, unauthorized usage, and attacks.

3. Managers and specialists overseeing peering management, or health of a total corporate telecommunications environment.

4. Managers and specialists responsible for information assurance and infrastructure protection, either for a specific Internet community (such as financial services, or the DoD networks), or for the entire Internet (such as CERT, NCS, NIPC, and other private and governmental organizations).

Software

GRADUS is a commercial service. See product website http://gradus.renesys.com or contact Renesys by email: support@renesys.com or by phone: (603) 640-6180.

Other None.

47 Centralized Secure Collection of Data from Multiple Sites

Contributors

Telcordia Technologies

Capability

Co-ordinated secure collection and storage of traffic measurements from probes at multiple sites under the administrative control of a central server at one of the sites, using host-based authentication in order to administer the probes without human intervention.

Usage

Software can be made available to NMS Program participants for (a) configuring the probes at the various sites for operation under centralized administrative control, and (b) configuring central server for trace collection and storage.

Software

Send e-mail to mcintosh@research.telcordia.com

Other

None

48 OSC Flow Compression

Contributors

Telcordia Technologies

Capability

On-line compression of network flow-data, specified in SAIC's OSC (On-line Simulation and Control) framework, to reduce bandwidth requirements for transport over network links, for real-time QoS performance analysis and display as well as for performance tuning on the basis of real-time simulation.

Usage

Integrate with OSC APIs

Software

C-code under development at Telcordia Technologies

Other

None