OMVis - A 3D Network Protocol Visualization Tool for OMNeT++

Harald Meyer
University of Vienna
Dep. of Distributed and
Multimedia Systems
Vienna, Austria
harald.meyer@univie.ac.at

Thomas Odaker
Johannes Kepler University
Linz
Inst. for Computer Architecture
Linz, Austria
todaker@gup.jku.at

Karin Anna Hummel
University of Vienna
Dep. of Distributed and
Multimedia Systems
Vienna, Austria
karin.hummel@univie.ac.at

ABSTRACT

Simulation of networks is a well-known way to develop, understand, and evaluate network protocols and algorithms. Standard tools for analyzing simulations often utilize aggregated statistics, histograms and other types of plots. With the growing interest in mobile networks and their specific characteristics like energy restrictions, limited bandwidth and shared usage of the transmission medium, the spatio-temporal behavior of algorithms gets more and more important.

In this paper we present OMVis, a complementary tool to traditional means of simulation analysis which focuses on the intuitive and spatio-temporal visualization of simulation traces. It provides 2D and 3D views to visualize node mobility, performance measurements, and properties of links between nodes. Multiple simulation runs and views can be shown in parallel for comparative protocol analysis and parameter studies. Various visualization cues, which may represent different simulation variables, offer convenient means to visualize a broad range of networks.

Categories and Subject Descriptors

I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

General Terms

Performance, Measurement, Verification, Design, Experimentation

Keywords

Visualization, OMNeT++, MANET, Routing, OpenGL

1. INTRODUCTION

When developing and evaluating network protocols by means of simulation, classic tools for analysis address aggregating networking observations and statistical evaluation. Basic visualization is provided by Matlab or R, or other tools capable of aggregating and visualizing data by using histograms, plotted time series, etc. These means are important and often sufficient to evaluate and compare

different approaches. However, statistical evaluation sometimes oversees rare or temporary phenomena. Thus, there is a need for additional support in the first steps of protocol implementation as well as for validation and to detect failures in the protocols used.

One way to approach this gap is to use an additional, flexible tool for intuitive protocol evaluation based on visualizing the major properties of network protocols. Our approach makes it possible to observe the behavior of the protocol dynamics over *time*, and, if mobile networks are considered, also in *space*. Hereby, anomalies and peculiarities, such as temporary full message buffers, can be detected by the protocol developer, even if they just occur transiently. These anomalies can be further analyzed using traditional tools together with the visualization tool.

In this paper, we describe our approach to ease protocol development by introducing OMVis, a platform independent visualization tool, as an extension to OMNeT++ [10]. OMVis is based on a definition of a basic set of network characteristics of interest together with appropriate means for visualization. Since we want this visualization tool to be open to new protocol demands, the set of network characteristics is extensible and the type of visualization can be selected by the protocol developer from a predefined, appropriate set of options. A major advantage and novelty of OMVis is the possibility to observe statistics not only over time, but also in different spatial areas in the whole simulated space.

Among the most important characteristics to be visualized for modern and future, mobility-assisting network protocols, are:

Network performance metrics. This category of characteristics contains classic performance metrics, such as (average) capacity, packet loss rate, or throughput of a link or the whole network.

Network node and link context. In addition, in mobile and dynamic networks, we see other network characteristics of utmost importance to validate network protocols. These characteristics describe the node and link states, which influence performance metrics and are important to understand the protocol's behavior. Such characteristics are, e.g., the message buffer size available, node degree (number of links to other nodes), energy consumed by nodes, and mobility behavior. When investigating ad-hoc networks, the node density within an area is an additional parameter of interest.

In the remaining paper, we will discuss related visualization tools and their capabilities and discuss the similarities to our tool in Section 2. To the best of our knowledge, this is the first visualization tool for OMNeT++ supporting the visualization of statistics over the time and space domain. In Section 3, we introduce OMVis and

its capabilities. Additionally, we demonstrate along selected examples, how OMVis can support protocol development and validation at early stages of development in Section 4. Moreover, we provide a qualitative discussion of the benefits of OMVis. Section 5 concludes the paper with an outlook on planned future developments.

2. RELATED WORK

Basic visualization of network statistics like line-, and box charts or histograms is natively supported by the OMNeT++ IDE, respectively by OMNeT++'s companion tools "plove" and "scalar". Common tools used for advanced analysis are GNUPlot[1] and Octave[2]. To the best of our knowledge there do not exist visualization tools for OMNeT++ which can give a global and intuitive overview over the spatial and temporal behavior of simulation traces. In the remainder of this section we will thus summarize related major visualization tools developed for other simulators.

A variety of visualization tools have been developed for ns-2 which itself provides Network Animator[3] (NAM). NAM was originally developed for wired networks and later extended for wireless simulations and provides only basic means of visualization. iNSpect [6] was designed especially for ns-2 traces of wireless networks. Nodes are drawn on a two dimensional plane. When a packet is transmitted over multiple hops, a path is drawn by interconnecting lines between each transmitting node and stays alive under node movement. Further different node colors provide means to analyze different node states during transmission. YAVISTA [1] focuses on the visualization of IEEE 802.11 packet schedules. It allows step-by-step analysis of packets with exact transmission times and logical dependence between packets. Huginn [9] is a ns-2 visualization tool very similar to OMVis. It offers 3D visualization of two dimensional MANET data. Nodes move on a 2D plane and the third dimension can be used for statistics, e.g., visualized by bars. Other visualization clues include circles that indicate each node's transmission range and text annotations. In contrast to Huginn, OMVis supports also stacked bars where different variables can be visualized in a single bar, visualization of mobility traces, and links between nodes. Moreover, it offers histogram views.

3. OMVIS OVERVIEW

OMVis is an OpenGL-based 3D visualization tool for OMNeT++ simulation traces. It is implemented in C++ and provides a GTK based user interface which allows real-time user interaction and visualization configuration. From a development point of view a focus was set on platform independence and POSIX compatibility. As a result, OMVis uses only cross-platform libraries like DevIL, OpenGL, GTK, zLib and TinyXML. It was successfully tested under various Linux and Windows platforms.

Simulation data can be visualized by histograms, boxes, discs, cones, lines, and colors. The developer can assign simulation data arbitrarily to each visualization type. Histograms provide an overview over current or aggregated measurements which could for instance be useful when analyzing each node's workload of a load balancing algorithm. Boxes are used to represent the measurement value, like the current packet queue size at a node. Discs visualize the ratio of the current value to the maximum value of a simulation variable. For instance, if the current workload is 25 percent, then a quarter of the disc is filled. Additionally, discs are used to visualize positions of nodes and special static entities like wireless

access points. Cones are used to represent special states of nodes, like nodes that are cluster heads. The thickness and color of lines is used to represent link-specific performance measurements, like the current bandwidth. Finally, color is used to visualize the ratio of the current value of a variable to the maximum value of the variable during the simulation. For instance, if the color of a box is green, then its associated variable is low compared to the maximum value. The more the current value increases, the more the color of the box cross-fades towards red.

Network simulations can cover a wide range of different measurements like for instance packet rates, energy usage or workload to name some examples. OMVis supports different measurements by allowing fully generic assignment of measurement variables to all visualization types and coordinate axes. For example, the x- and y-axis could be assigned classical geographic location information or network coordinates [7]. Further variables can be connected by expressions like addition and multiplication and the use of constants for each axis is possible. For instance, the height of a box could represent the sum of unicast and broadcast packets where each, unicast and broadcast packets are different variables and thus are stored separately in the simulation trace file.

Simulations can be replayed with different speeds, and a time-slider allows jumping within the simulation. Different visualization cues can be switched on and off in real-time. The OpenGL view supports full mouse interaction including movements, rotations and zooming. OMVis allows to present multiple simulation views in parallel only limited by memory capabilities and screen size (see, e.g., Figure 2). The simulation runs can be linked together. Hence, different parameter values at the same simulation time can be compared or each simulation view can run independently (e.g., for comparative algorithm studies). It is even possible to combine the variables of multiple simulation runs within one view.

OMVis supports four different views (see Figure 1): a *3D node view* (bottom left), a *3D link view* (top left), a *2D histogram view* (top right), and a *2D cumulative histogram view* (bottom right). The 3D node view shows the general network topology together with the selected variables for visualization at a given time. The 3D link view shows direct activity between distinct nodes, like the number of packets transmitted. Link information is separated from the 3D view because the sometimes huge number of nodes, and the nodes' mobility would visually overload the 3D node view. The 2D histogram view shows a standard histogram of a selected variable at the current simulation time, while the cumulated histogram draws the variable's values over the complete simulation timespan. In the following Sections each view is presented in detail.

3D Node View

In the 3D node view, simulation data is visualized on a plane in 3D space. This plane could for instance represent the network's playground. Beside the visualization of positions on the plane this view supports height information (visualized by boxes), color and discs which float above the nodes. The height of boxes can express single or multiple variables. For each variable a different color can be selected. If no color is assigned, the color is blended from green to red, depending on the ratio between the current measurement value to the maximum measurement value within the whole simulation. The discs can be used to represent the ratio of the current measurement value to the maximum measurement value. For example, if the current load is 25 percent (maximum load is 100 percent), a quarter of the disc is filled. Aggregated mobility of nodes can be visualized by a map which is drawn on the plane. A color cue is used to represent density on a given point over the simulation time.
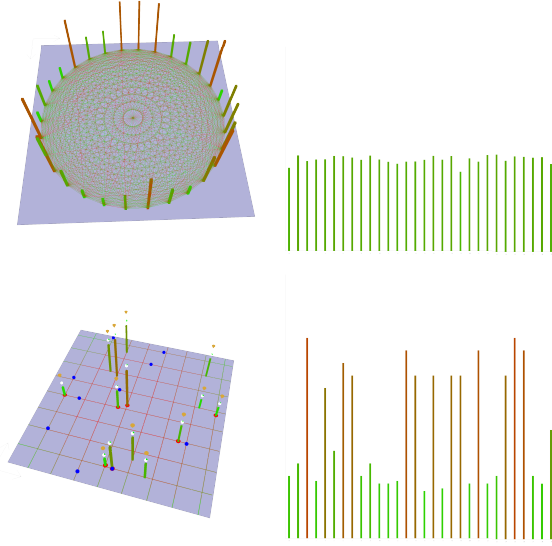
Figure 1: The four views of OMVis.

OMVis also supports visualization of fixed, non-moving objects, like wireless access points. Such objects are drawn in a different color.

3D Link View

The 3D link view shows nodes arranged on a circle at fixed positions which are interconnected by links. Each node is represented by a box. The height of a box could for instance represent a local queue size. The usage of a link is represented by color and line thickness and could represent signal to noise ratios or the bandwidth.

2D Histogram View

In the 2D histogram view, a standard histogram is drawn. Each node has a fixed position (depending on its index). The height of each bar represents an user selected measurement variable's value. The visualization is supported by a color cue which blends from green to red, depending on the ratio of the current value to the maximum value within the whole simulation.

2D Cumulative Histogram View

The 2D cumulative histogram view is a fixed view which shows the aggregated/cumulated values over the entire timespan of the simulation.

Additional Features

OMVis trace files can cover a wide range of measurement variables. To speed up loading and to reduce memory consumption the measurement variables to be loaded have to be specified in a control file which is parsed when OMVis is started.

A problem that arose during the implementation of OMVis was the potentially large size of the source data. This caused memory problems on machines with small amounts of RAM. In order to reduce this problem data can be stored in compressed form in memory. When specific data is required it is decompressed on-the-fly in memory using `zlib`. On a state-of-the-art Intel Core2Quad 2.5GHz machine with 4GB DDR2 RAM, decompression of 1GB of data takes between 1 to 2 seconds (only one CPU core is used

by zlib). Compression, which is done only once at startup is more time-consuming. For instance the compression of a random 1GB sample took about 43 seconds on average on the same machine.

Another feature is to filter the nodes that are drawn. A filter basically specifies an expression, a comparison, and a value to compare to. If the specified condition is met, the node is ignored when drawing. Additionally it is possible to specify that the nodes that are selected by the condition are highlighted instead of filtered to aid the user in identifying nodes that meet some condition. A possible filter could be, for instance, "show all nodes which hold five replicas and have between 40 to 60 percent of energy left".
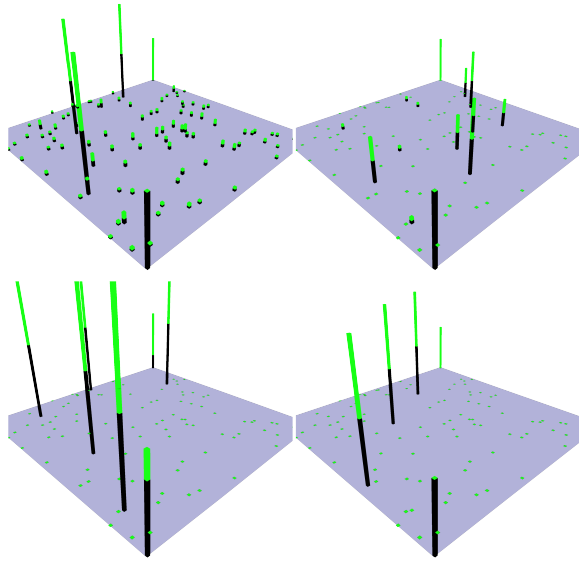
4. EXPERIMENTS

In this section we present two examples where OMVis can support simulation analysis and understanding. One example demonstrates how the spatio-temporal visualization of the number of packets generated on each node can support understanding of MANET routing protocols. The visualization provides, e.g., intuitive means to see frequent routes. The second example shows how the visualization of mobility traces can be used to find spatio-temporal model errors or programming bugs which might not be visible in statistics.

In the first example the number of unicast packets sent and received by each node using different MANET routing protocols is visualized. Traffic generated for acknowledgments and broadcasting is omitted for reasons of clarity.

We selected routing protocols which are already implemented in the INETManet framework [3], which are Ad hoc On-Demand Distance Vector (AODV) routing [8], Dynamic MANET On-Demand (DYMO) routing [2], Optimized Link State Routing (OLSR) [4], and Dynamic Source Routing (DSR) [5]. OLSR is a proactive protocol which maintains routes independent of demand, while the other three protocols are reactive and build up/fix routes only when payload has to be routed. The simulation was conducted in a playground of 1000m by 1000m with 100 static nodes. One node was placed on the lower right of the playground and sent UDP packets with a rate of 10 packets/second to another node placed on the top left of the playground. Both nodes were not in each others transmission range. The remaining nodes were placed uniformly random on the playground. The network was based on 802.11g ad hoc mode with 54MBit data rate. Routing parameters were set to INETManet's *default* values.

In Figure 2, the number of unicast packets sent (lower part of the bar) and received (top part of the bar) by each node after 100 seconds are depicted. The 3D visualization reveals the different load distribution among nodes and allows to intuitively guess transmission routes. By using the interactive time-slider, the transmission path of each packet can be tracked by watching the bars grow. Further similarities between the selection of routes by different protocols can be seen (e.g., AODV vs. DYMO).

The second example shows the visualization of mobility traces. In this example nodes move according to the Manhattan mobility model. Here, each node moves along predefined streets. When a node arrives at an intersection it turns left or right, reverses, or does not change its direction of movement with a given probability. When a node reaches the playground's border its moving direction is inverted. In Figure 3 movement traces of a sample simulation are depicted by lines drawn on the floor. It can be seen that the movement traces are sometimes faulty, e.g., nodes leave the predefined streets. This error might probably not have been discovered without the visualization. By using the time-slider the exact times when the error occurs can be obtained and used for further evaluation.

Figure 2: Number of unicast packets sent (lower part of the bars) and received (top part of the bars) for AODV (top left), OLSR (top right), DSR (lower left) and DYMO (lower right) MANET routing protocols.

Figure 3: Visualization of node's mobility traces, special states and fixed points.

Discussion of Experiments

The two presented examples clearly show the advantages of the three dimensional visualization approach of simulation data. The 3D node view allows to quickly discover the spatio-temporal distribution of simulation variables' values - like the number of sent and received packets by observing the height of boxes which represent the packets sent and received. The multi-3D-view approach provides algorithm designers a tool to compare different algorithms or different parameter studies at a glance over time and space. The 2D histogram views can be used to observe the distribution of variables, e.g., the packets sent or received, at the current simulation time and over the complete simulation timespan. The 3D link view can be used to visualize the load between distinct nodes, and thus helps to find out if distinct nodes excessively communicate. Further the aggregated temporal visualization of, for instance, mobility traces can lead to the detection of bugs of parts of the simulation that are thought of to be correct, such as the bug presented in the Manhattan mobility model where the accumulated mobility traces drawn on the plane show incorrect paths.

5. CONCLUSION AND FUTURE WORK

We presented OMVis, a platform independent visualization tool for OMNeT++ simulation traces. Multiple 3D and 2D views offer means for visualization of node topologies, link related data and performance measurements. Cues for visualization include boxes, discs, lines, cones, and colors. The assignment of simulation variables to cues can be fully controlled by the user.

OMVis provides an intuitive spatio-temporal access to simulation traces, but it should not be seen as a replacement for statistics. Only the complementary use of both, OMVis and statistics provides useful means for algorithm understanding and analysis. For future work it would be useful to integrate statistics into OMVis. Further we think of integrating means for automated analysis, like mechanism to detect and highlight heavy loaded regions within the network. It is planed to release an open source version of OMVis
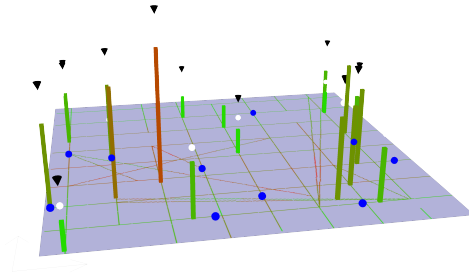
in the near future.

7. REFERENCES

[1] R. Ben-El-Kezadri and F. Kamoun. YAVISTA: A Graphical Tool for Comparing 802.11 Simulators. *JCP*, pages 10–20, 2008.

[2] I. Chakeres and C. E. Perkins. Dynamic MANET On-Demand (DYMO) Routing. Internet-Draft, 2006.

[3] INETManet Framework. Version 18th November 2009, `http://github.com/inetmanet`.

[4] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol. In *INMIC '01*, pages 62–68, 2001.

[5] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[6] S. Kurkowski, T. Camp, N. Mushell, and M. Colagrosso. A Visualization and Analysis Tool for NS-2 Wireless Simulations: iNSpect. In *MASCOTS '05*, pages 503–506, 2005.

[7] T. Ng and H. Zhang. Predicting Internet Network Distance with Coordinates-based Approaches. In *INFOCOM '02*, pages 170–179, 2002.

[8] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *MCSA '99*, pages 90–100, 1999.

[9] B. Scheuermann, H. Füßler, M. Transier, M. Busse, M. Mauve, and W. Effelsberg. Huginn: A 3D Visualizer for Wireless ns-2 Traces. In *MSWiM '05*, pages 143–150, 2005.

[10] A. Varga and R. Hornig. An Overview of the OMNET++ Simulation Environment. In *Simutools'08*, pages 1–10, 2008.