Animations for a Computer Networks Course - Practice and Experience

Author:

Rainer Oechsle, University of Applied Sciences, D-54208 Trier, Germany, oechsle@informatik.fh-trier.de

Abstract — *In this paper we will present our approach of enhancing a traditional course on Computer Networks by animations. This course is part of the bachelor programme in computer science at the University of Applied Sciences, Trier, Germany. It is well-known that the effort for constructing a completely self-contained teaching unit that is suited for self-studying is tremendous. A teacher standing in a class can react in a very flexible way to questions that show that a certain aspect has not been covered well enough in order to be understood by all the students. Teaching software must anticipate all these problems and offer adequate help which is not easy to do. Therefore our goal was less ambitious. We tried to identify certain mechanisms within the area of computer networks that are well suited for visualizations. Candidates for animations had to be sufficiently complex. It is not worthwhile, or sometimes even boring, to present easy-to-understand subjects with animations. The animations that we built are used in an interactive way which guarantees high flexibility. This means that the user can and must decide before each step what will happen next. It is important that each step is simple enough. If by pressing a button many things happen at the same time, it is impossible to follow and understand what is going on there. Moreover, we believe that even the few changes of a scene had to be done with smooth transitions instead of immediate modifications. Otherwise the changes will be hard to track for human beings. Finally, we are convinced that one of the biggest advantages of building animations is more on the developers side than on the viewers side. Students have fun in developing animations because they get immediate visual feedback from their work. In addition, they acquire a more thorough understanding of the mechanisms to be visualized than by conventional treatment. This paper contains some examples of our animations (traffic flow in virtual local area networks, TCP protocol, routing protocols, visual sniffing tool) and describe our experience in using them.*

Index Terms — *Animations, Computer Networks, Routing Protocol, Sniffer, TCP, Visualization, VLAN.*

INTRODUCTION

Within the last years interest in computer-based training (e-learning) has increased. It is well-known that the effort for constructing a completely self-contained teaching unit that is suited for self-studying is tremendous. A teacher standing in a class can react in a very flexible way to questions that show that a certain aspect has not been covered well enough in order to be understood by all the students. Teaching software must anticipate all these problems and offer adequate help which is not easy to do. Therefore our goals were less ambitious:

- We selectively identified certain mechanisms within the area of computer networks that are well suited for visualizations. Candidates had to be sufficiently complex. It is not worthwhile, or sometimes even boring, to present easy-to-understand subjects with animations.
- The animations that we built are used in an interactive way which guarantees high flexibility. This means that the user can and must decide before each step what will happen next. The animations therefore have to incorporate a big part of the logic of the visualized mechanims. It is thus not sufficient to have a tool that just steps through certain prefedined snapshots.
- It is important that each step is simple enough. If by pressing a button many things happen at the same time, it is impossible to follow and understand what is going on there.
- We believe that even the few changes of a scene had to be done with smooth transitions instead of immediate modifications. Otherwise the changes will be hard to track for human beings.
- We do not claim that our animations are suited for self-studying. Instead , the animations are used as supporting material in traditional lectures. After the lecture students may download the animations onto their PCs and repeat the examples given in the lecture or try out new ones.

The outline of the paper is as follows. In the main part of this paper four examples of animations are presented. The examples cover different key areas of computer networks education [1]-[3]:

- traffic flow in vlans (virtual local area networks)

- connection setup and teardown, reliability and flow control mechanisms of the TCP protocol
- a distance-vector algorithm for computing routing tables
- visualization of captured data

More technical details can be found in [4]-[6].

TRAFFIC FLOW IN VIRTUAL LOCAL AREA NETWORKS (VLANS)

Local area networks today are typically switch-based Ethernets. Network administrators have the possibility to define virtual local area networks (vlans) and logically connect the PCs to one ore more of these vlans. Only the PCs within a vlan can communicate directly with each other. Communication between PCs connected to different vlans is only possible with the help of one or more IP routers which forward the data packets. Because the logical configuration of vlans and PCs is totally independent of the physical configuration, it is not obvious what is happening when two PCs communicate.

Therefore we built a vlan simulator that offers the possibility to define one or more vlans, logically connect PCs and routers to one or more of the configured vlans, and observe the sending of data from one PC to another crossing several Ethernet switches and IP routers.

Figure 1 contains a screenshot of our vlan simulator. The panel on the left contains the physical configuration consisting of host PCs, routers, and Ethernet switches. The currently active devices are highlighted. In the lower part a message flow diagram shows all the messages that have been sent and received. The upper right panel contains the logical view which shows how PCs and routers are attached to the different vlans. Below this logical view there is a textual description of what is happening at the moment. The lower right part contains lists to select the sending and receiving PCs and buttons to control the progress of the animation.

KEY MECHANISMS OF THE TCP TRANSPORT PROTCOL

The forwarding of data packets in the Internet is unreliable: packets may be lost or delivered out of order on the receiving side. The TCP protocol (Transmission Control Protocol) is a rather complex protocol that offers reliable data transport in the Internet. Its main characteristics are:

- The TCP protocl is connection-oriented. Thus two partners habe to setup a connection before being able to exchange application data. At the end the connection is closed.
- The TCP protocol is reliable. Reception of data is acknowledged by the receiver. If the sender does not get an acknowledgement after a certain amount of time, the data is sent once more.
- The TCP protocol contains a flow control mechanism. This means that the receiver allows the sender to send a certain amount of data thus preventing the receiver being overflown by the sender.
- The TCP protocol is data-stream-oriented. The application on the receiving send is able to receive data in portions that are independent of the portions of the data sent.

We built a visualization of the TCP protocol that animates the described main features of this transport protocol. In order to keep the simulation simple, application data can only be sent from the left to the right. Acknowledgements flow in the opposite direction. Figure 2 presents a screenshot of our animation program. The lower part of the window serves as a control panel for the highly interactive use of the animation program. The upper part shows the state of the simulation. The left side shows the sender with the data that has been sent but not yet acknowledged in red and the data that the sender is allowed to send according to the flow control mechanism in green. The upper middle part shows a message sequence diagram of all sent and received messages. Data currently in transit is visualized in the lower middle part. The receiver is shown on the right.

A DISTANCE-VECTOR ALGORITHM FOR COMPUTING ROUTING TABLES

IP routers forward data packets. The decision on which outgoing port the data packet is going to be sent is based on the router's routing table. These routing tables are not configured in a static way, but are continously updated to react to changing network situations. There are two algorithms for the distributed computation of routing tables: link-state algorithms and distance-vector algorithms.

In distance-vector algorithms a router sends his routing table to all its immediate neighbors. On receiving such a message a router updates its routing table. In case the routing table was changed by the update, the router sends its updated table to all its neighbors. This process continues until there are no more changes. All the router's routing tables are computed.

Figure 3 shows a screenshot of our animated simulation of such a distance-vector algorithm. The main window contains the network topology that the user can interactively create. Below the main window there are six windows for each of the routers in the topology window. Each of these windows contains the data of the last received routing table, the current routing table (i.e. the routing table after the update triggered by the received routing table), and the routing table before the update. It is thus easy to study the modifications of all routing tables. Routers that received a message during the last simulation step show the text in red. The window on the lower right titled Simulation serves as the control panel for the simulation.

VISUALIZATION OF CAPTURED DATA TRAFFIC

What we have seen so far are simulations. In order to convince students that there is no cheating in these simulations, we offer a tool that is able to capture real data traffic that is sent and received over the network adapter of the student's PC, and that visualizes the captured data packets. Thus students observe reality instead of simulations.

There a quite a few tools around for capturing and showing network traffic. In [6] we describe why we believe that these tools are not suited for computer networks education purposes. The main goal of our tool called VisuSniff is the presentation of captured network traffic in a graphical form that is especially suited for computer science education. After data has been captured, an overview diagram shows all captured communication relations. By clicking on specific items in the overview diagram (hosts, ports, communication relations), the interesting information can be selected easily. A message sequence diagram then visualizes all exchanged data packets chronologically for the selected items (see the right part in Figure 4). After clicking on a message flow arrow in the message sequence diagram, a visual representation of the exchanged data is shown. The data is presented in a way which is typically used in lectures and text books (see the two windows on the left in Figure 4 with more and less details).

EXPERIENCE

The experience that we gained during the development and usage of the described and other simulations, can be summarized as follows:

- The mechanisms to be visualized should be carefully selected. Sufficiently complex mechanisms are well suited. If the lecture is filled with animations even for facts that are simple to understand, students get tired with these animations. The animations should really be used as highlights. They should only be used in case that traditional media like overhead projectors and blackboards reach their limits. If animations are incorporated in the way just described, we get extremely positive feedback from our students.
- Animations should be actively controlled by the users. An animation should not only be passively observed like a movie. This motivates the users to acquire the basic knowledge in order to be able to use the animations at all. If students ask questions like "What will happen if ...?", it is easy to answer the question just by watching the simulation.
- Finally, we are convinced that one of the biggest advantages of building animations is more on the developers side than on the viewers side. Students acquire a more thorough understanding of the mechanisms to be visualized than by conventional treatment. Repeatedly, we had to look up details that are usually not described in computer networks text books [1]-[3]. In addition, students have fun in developing animations because they get immediate visual feedback from their work.

ACKNOWLEDGEMENT

I would like to thank the following people for their design and implementation work: Hermann Schloß implemented the vlan simulator, Thorsten Martin and Thomas Schmitt realized the TCP simulation, Bernd Greve and Christoph Mettler developed the distance-vector algorithm simulation, Oliver Gronz and Markus Schüler programmed the visual sniffer VisuSniff.

REFERENCES

[1] Kurose, J., Ross, K., "Comuter Networking", Addison-Wesley, 2000

[2] Peterson, L.L., Davie, B.S., "Computer Networks - A Systems Approach", Morgan Kaufmann, 2000

[3] Tanenbaum, A.S., "Computer Networks", Prentice-Hall, 1996

[4] Oechsle, R., Becker, R., "Das Projekt Vivaldi (Visualisierung von ausgewählten Lehrinhalten der Informatik)", *Architektur von Rechensystemen – Systemarchitektur auf dem Weg ins 3. Jahrtausend: Neue Strukturen, Konzepte, Verfahren und Bewertungsmethoden – Vorträge der 15. GI/ITG-Fachtagung ARCS '99 und der APS '99 vom 4. bis 7. Oktober 1999 in Jena*, 1999, pp. 263-270 (in German)

[5] Oechsle, R., Jahn, D., "Visualisierung von ausgewählten Lehrinhalten der Informatik – Beispielanimationen aus dem Projekt Vivaldi", *Softwarevisualisierung 2000, Schloß Dagstuhl, Germany*, May 2000, pp. 79-90 (in German)

[6] Oechsle, R., Gronz, O., Schüler, M., "VisuSniff: A Tool for the Visualization of Network Traffic", *Proceedings of the Second ACM Program Visualization Workshop, Hornstrup Centret, Denmark,* June 2002, pp. 118-124

FIGURES AND TABLES

FIGURE 1

SCREENSHOT OF THE VLAN SIMULATOR.

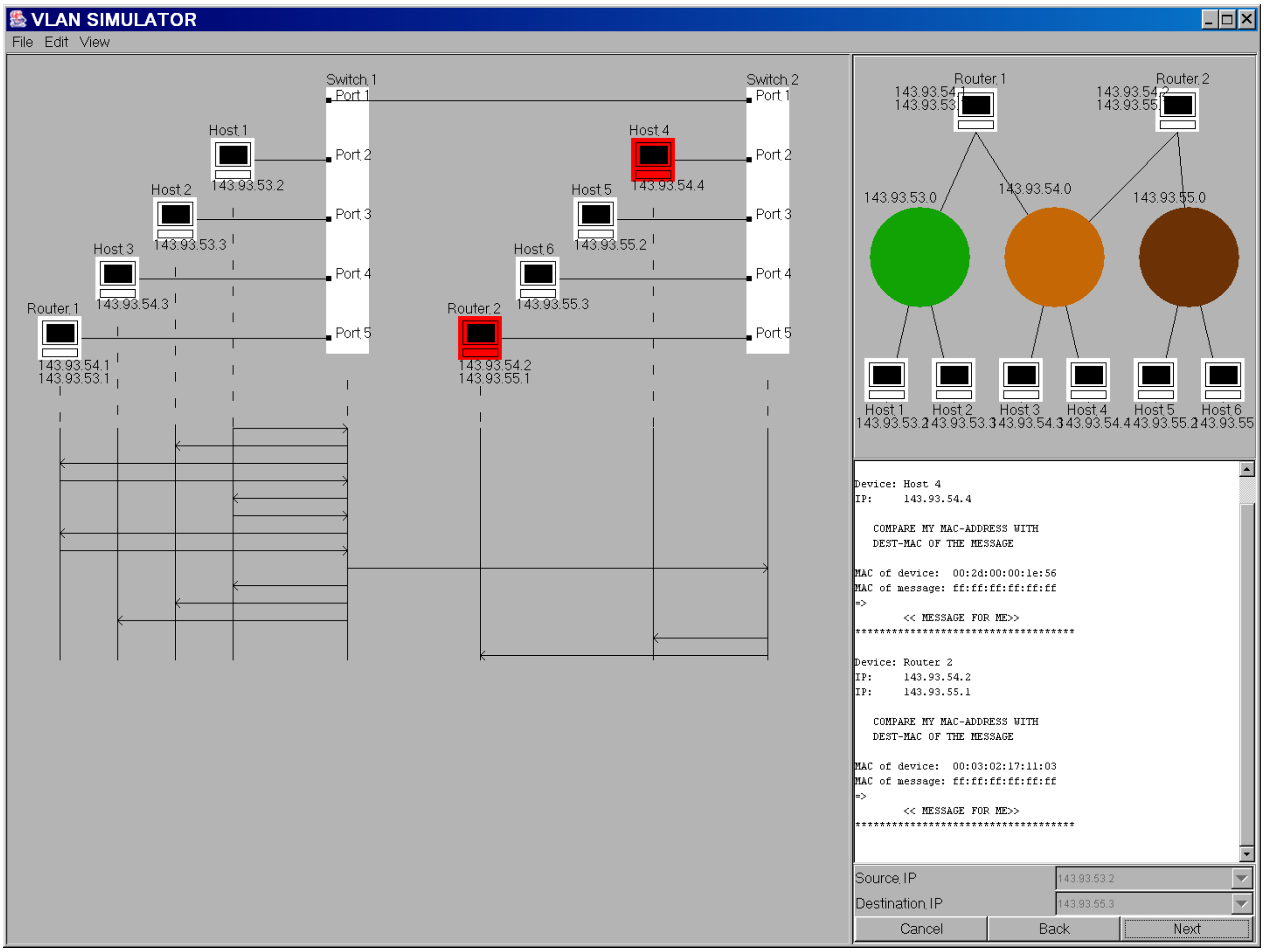

FIGURE 2
SCREENSHOT OF THE TCP ANIMATION.

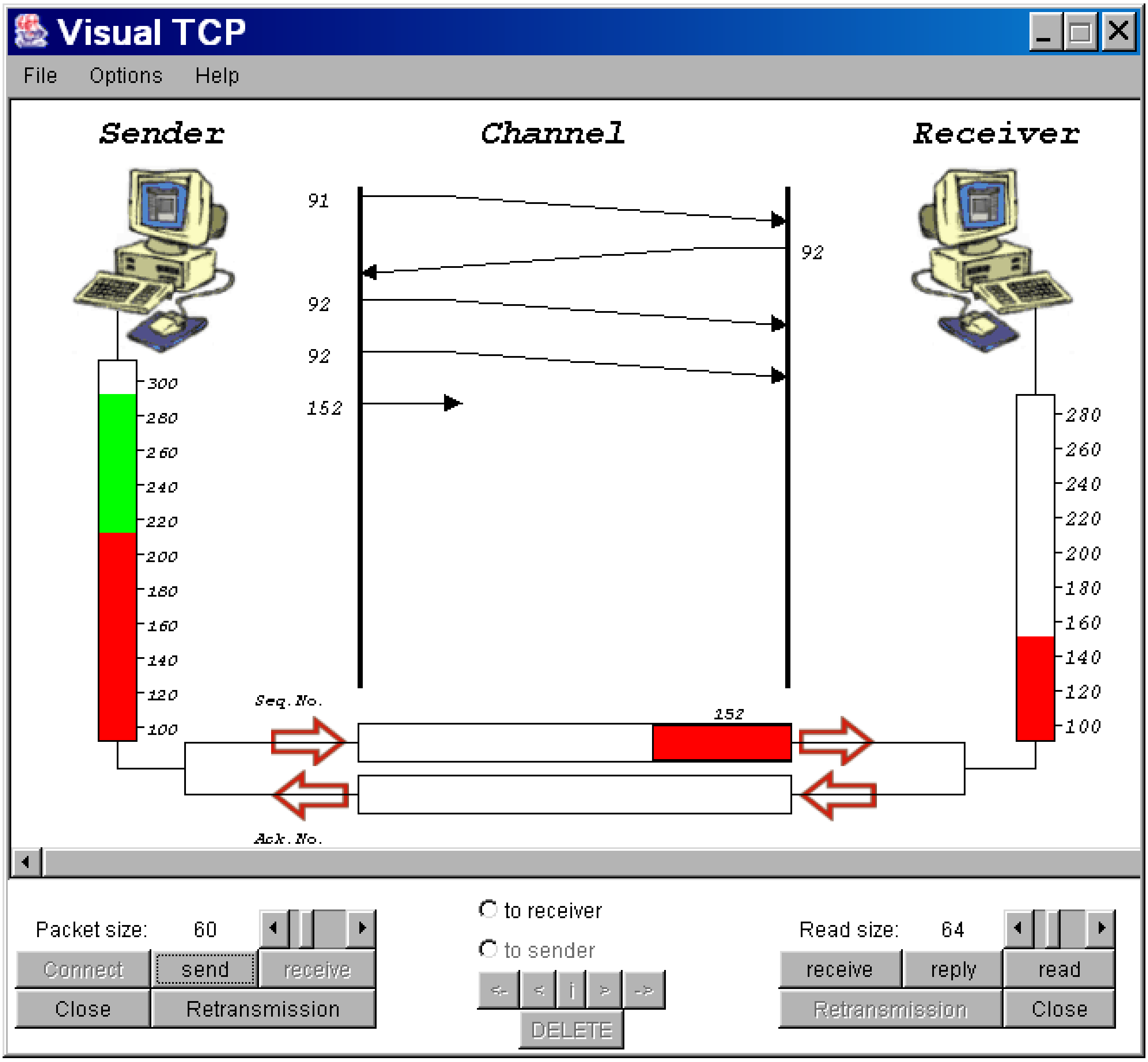

FIGURE 3
SCREENSHOT OF THE ANIMATED SIMULATION OF A DISTANCE-VECTOR-ALGORITHM.

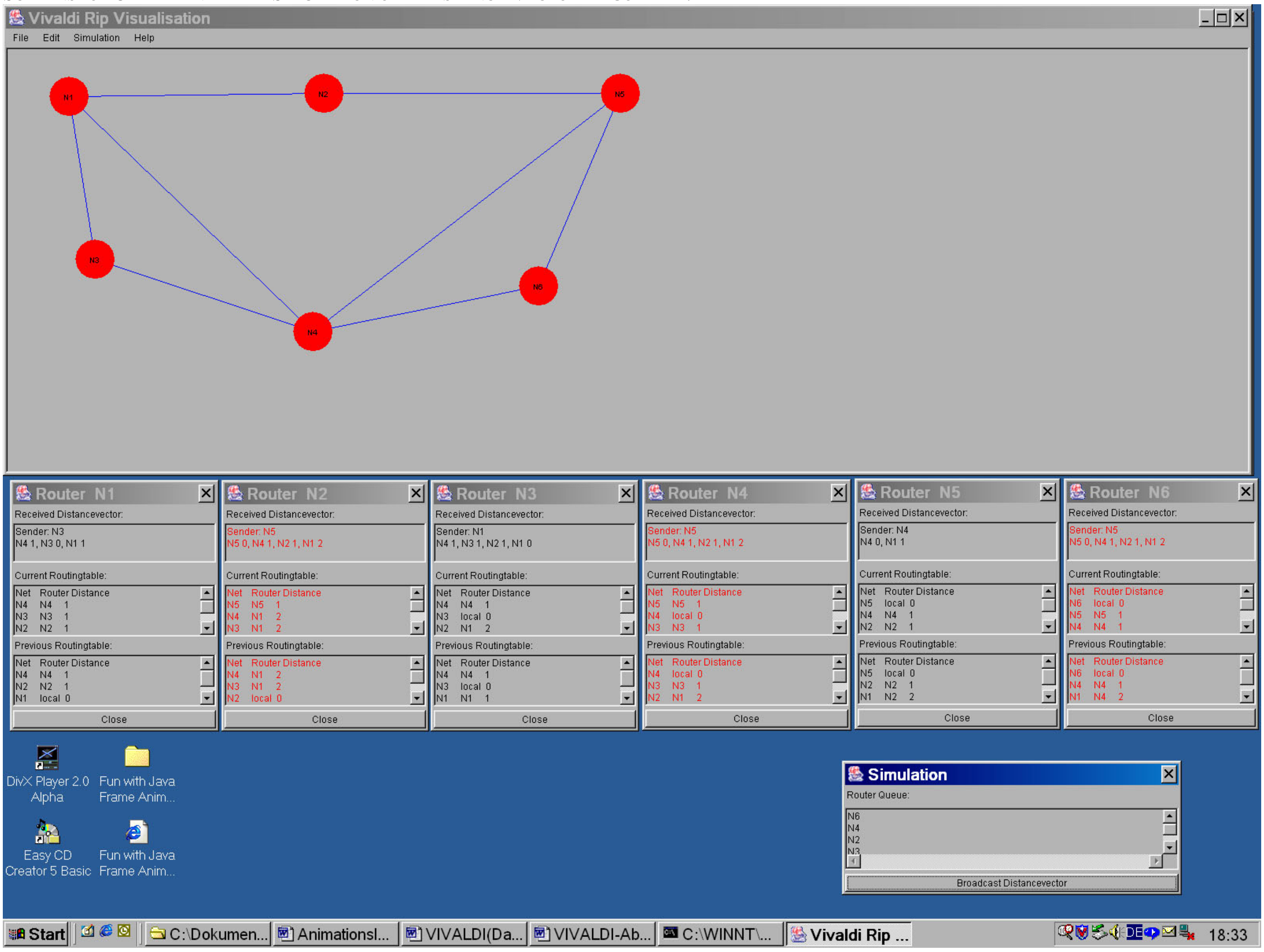

FIGURE 4
SCREENSHOT OF VISUSNIFF.