Visualizing Network Data

Richard A. Becker, Stephen G. Eick[†], and Allan R. Wilks

***Abstract*—Networks are critical to modern society, and a thorough understanding of how they behave is crucial to their efficient operation. Fortunately, data on networks is plentiful; by visualizing this data, it is possible to greatly improve our understanding. Our focus is on visualizing the data associated with a network and not on simply visualizing the structure of the network itself. We begin with three static network displays; two of these use geographical relationships, while the third is a matrix arrangement that gives equal emphasis to all network links. Static displays can be swamped with large amounts of data; hence we introduce direct-manipulation techniques that permit the graphs to continue to reveal relationships in the context of much more data. In effect, the static displays are parameterized so that interesting views may easily be discovered interactively. The software to carry out this network visualization is called SeeNet.**

***Index Terms*—Network visualization, parameter focusing, network data analysis, interactive graphics, data analysis, direct manipulation.**

I. Introduction

We are currently in the midst of a networking revolution. Data communications networks such as the Internet now connect millions of computers, cellular phones have become commonplace, and personal communications networks are in the developmental stages. In parallel with the ever increasing network sizes has been a concomitant increase in the collection of network measurement data. Understanding this data is of crucial importance as we move to a modern, information-rich society.

Unfortunately, tools for analyzing network data have not kept pace with the data volumes. More network measurement data is available today than ever before, yet it is useless until it is understood. Traditional network analysis software and graphs cannot cope with the size of today's networks and their data collection capabilities.

We have developed some novel graphical tools for displaying network data, together with display manipulation techniques that can help extract meaningful insights from the masses of network data currently available. In addition, our tools scale to the even larger networks that are emerging.

A network consists of nodes, links, and possibly spatial information. Statistics, which may be raw data or data summaries and may vary over time, are associated with the nodes and the links. The link statistics may be directed, as in call flow of a circuit-switched network, or undirected, as in the network's capacity. The network may have a natural spatial layout as does a geographical trade-flow network, or may be abstract as in a personal communications network. Network data may be categorical, such as the type of node or link, or quantitative, such as a link's capacity. The data may be static, such as a network's capacity, or time varying, such as the network flow in several time periods.

Our focus is on visualizing network data and our motivating examples involve communications networks. Our goal is to understand the data and not the networks themselves. Thus the structure and connectivity of the graph are of secondary importance—data associated with the links and nodes is what we want to understand. For communications networks the important questions involve the sizes of the flows, the capacities of the links, link and node utilization, and variations through time.

A challenge in visualizing network data is coping with the data volumes—networks may have hundreds or thousands of nodes, thousands or tens of thousands of links, and data from many time periods. The three methods traditionally used to reduce the amount of network data to manageable size are:

- for large numbers of links or nodes, aggregation,
- for large numbers of time periods, averaging, and
- for detecting changes, thresholding and exception reporting.

Each of these data-reduction methods may obscure important information. Graphical displays, particularly network maps, have long been recognized as a crucial tool for analyzing network data. Graphical displays, however, even when running on powerful workstation-based systems, can become swamped and overly busy when faced with large networks and data volumes. We present techniques to address these display clutter problems.

We introduce three graphical tools, collectively named *SeeNet*[1], for visualizing network data. Our visualization techniques involve static displays, interactive controls, and animation. In Sections II, III, and IV we will illustrate these three techniques by using as an example the AT&T Long Distance Network—the national circuit-switched telecommunications network used by AT&T to carry long distance telephone calls. The example involves data from over 110 nodes (switches) that are (nearly) completely connected. Each node has a geographic location and there are statistics for each node, directed link statistics between every pair of nodes, and new data collected every five minutes. The challenge with this example involves the large number of links, over 12,000, and the time varying aspects of the data. Our data is from October 17, 1989, when an earthquake occurred in the San Francisco Bay area. In Section V, we will present our techniques using data from a

[†]Correspondence contact: AT&T Bell Laboratories - RM IHC 1G-351, 1000 E. Warrenville Road, Naperville, IL 60565 USA; e-mail: eick@research.att.com.

IEEECS Log Number V95003.

[1] It's a name, not an acronym; it lets you See a Network.

1077-2626/95$04.00 © 1995 IEEE

variety of networks, including Internet packet traffic and electronic mail messages within a research department. Section VI discusses SeeNet's uses within AT&T and describes its implementation, while Section VII places the work in the context of related research.

II. Static Network Displays

We now turn to a set of static techniques for displaying network data using telecommunications traffic among the 110 switches in the AT&T network on October 17, 1989, the day of the San Francisco earthquake. During a major network event such as an earthquake, interesting questions concern the network capacity and traffic flows:

- Where are the overloads?
- Which links are carrying the most traffic?
- Was there network damage?
- Are there any pockets of underutilized network capacity?
- Is the overload increasing or decreasing?
- Are calls into the affected area completing or are they being blocked elsewhere in the network?

The widely used techniques for statically displaying network link data involve displaying the data spatially on a map or representing the data as a matrix.

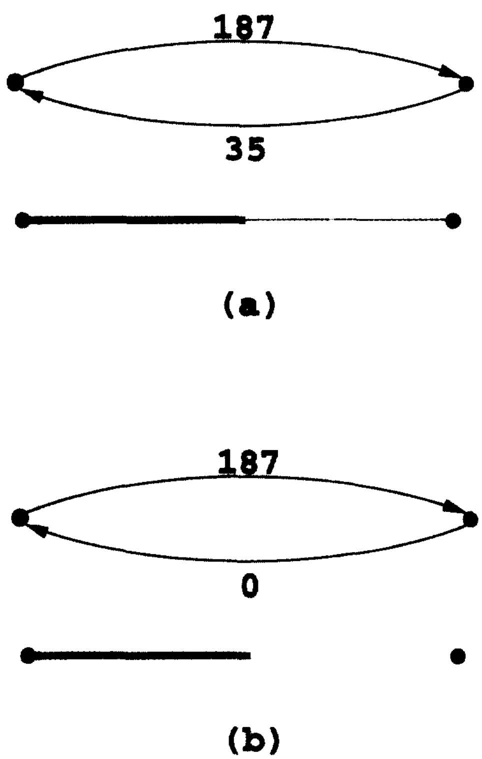

Fig. 1. Representing Link Data. In part (a), the upper illustration is a conventional arrow diagram with numbers to show link statistics; the lower illustration uses line thickness and color to convey the same information more compactly. Part (b) shows how half-lines represent statistics with value zero.

A. Link Maps

One way of displaying link data on a map is to draw line segments between each pair of nodes for which there is data. This pictures the connectivity of the network. To show the values of a link statistic, the segments may be colored, for example, or drawn with varying thicknesses. Since link data is often directional, it is possible to show data in both directions using arrows or by bisecting the segments and always using the half connected to a node to show the statistic with that node as the originating node.

Fig. 1 shows schematically how these segments are constructed. The upper half of part (a) shows that between the two nodes is a link with data for both directions; a compact form of this is represented in the lower half of part (a) by a pair of connecting line segments, with gray shade and thickness proportional to the data values. Part (b) of Fig. 1 illustrates a refinement that we often use to reduce clutter—if one of the data values is zero, the corresponding line segment is not drawn. A negative data value for a particular link can be shown using a dashed instead of a solid line.

Fig. 2 illustrates some of these ideas by showing a *linkmap* of the overload into and out of the Oakland node using only the bisected segments. The "island" on the map in the Atlantic Ocean is a blowup of the New York/New Jersey area, necessitated because of the larger density of nodes in that region. The message of this figure is clear; there is an overload into Oakland from every other node and out of Oakland to many of the nodes, particularly on the East Coast. The most heavily overloaded links are from Seattle, Denver, and some of the major cities on the West Coast.

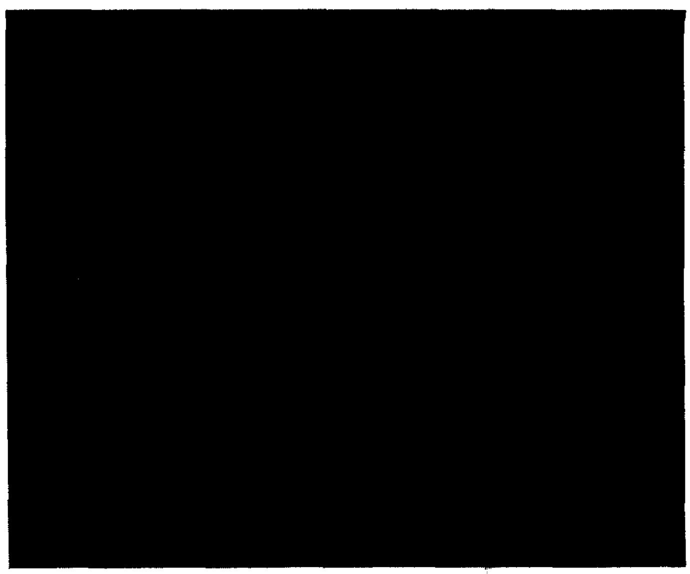

Fig. 2. Overload Into One Node. The overload into and out of the Oakland node. The half-lines between the nodes code the overload by direction.

The display in Fig. 2 works well because the lines encoding the link statistics do not overlap much. It is often the case, however, that these link-data displays have many overlapping lines and are therefore difficult to interpret. For example, look at Fig. 3, which shows the network-wide overload for the same time period as in Fig. 2. The plethora of lines makes the figure too complicated to understand easily. There is an overload focused on several of the nodes on the West Coast, but which of these nodes is most affected? Another difficulty is that the long transcontinental links cover the middle of the country and obscure potential problems underneath. A third problem is that it is hard to see where a half-line terminates if the other half is not displayed. Two simple techniques improve Fig. 3 by making the important links visually prominent. The important links (those with large overload) are drawn last so that they appear on top and are not obscured by the other links. The link thickness is used to encode the statistic redundantly, making the important links thicker and therefore more apparent.

Fig. 3. Node-to-Node Network Overload. The network-wide overload after the earthquake for the same time period as Fig. 2. The display is ineffective because there are many long lines that obscure much of the country.

A different solution to the map-clutter problem caused by drawing too many lines involves aggregation at each node, with the resulting node data displayed using a node-based map or *nodemap*.

B. Nodemaps

The idea embodied in a nodemap is to display node-oriented data by showing a glyph or symbol such as a circle or square at each node on the map, with the visual characteristics such as size, shape, and color of the glyph coding the value of the statistic. More complex symbols can be used to represent more than one statistic simultaneously. For example, if "call attempts" is the statistic under consideration, there will typically be a count of inbound calls and of outbound calls at each node; these can be shown by using rectangles for the symbols, with the width and height proportional to the two statistics.

Fig. 4 shows the nodemap representation of the AT&T network-calling statistics for the same period as in Figs. 2 and 3. The data shown in Fig. 4 is the aggregate overload into and out of each node, shown on a square root scale. The tall, thin rectangles indicate an outbound overload, the short, fat rectangles indicate an inbound overload, and the square rectangles indicate a symmetric load. The area of the rectangle is approximately proportional to the total overload because of the square root scale. There are clear and obvious geographical patterns. The overload is from virtually every node in the country primarily into the northern California nodes and secondarily into southern California. The heaviest overload is in California, on the West Coast, and decreases gradually west to east.

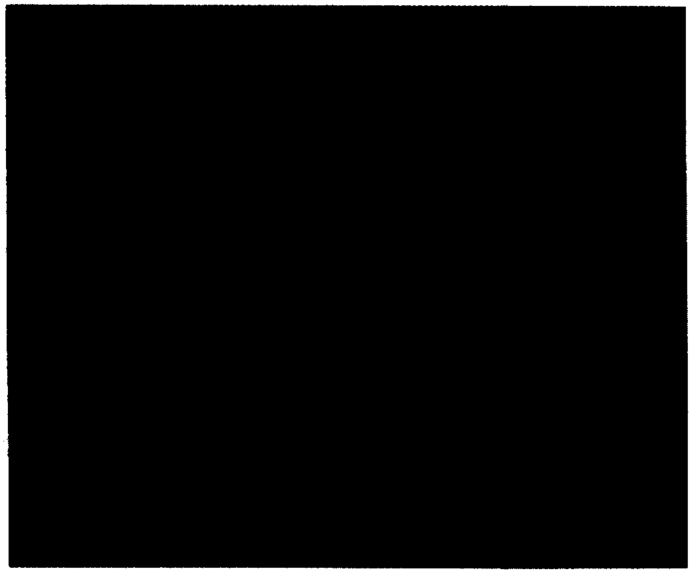

Fig. 4. Overload Following Earthquake. Each rectangle shows the aggregate overload over all links; its horizontal dimension is proportional to the square root of the number of incoming calls in the preceding 5-minute period, and the vertical dimension encodes the outgoing calls.

A nodemap solves the display clutter problem in Fig. 3 nicely via aggregation. Unfortunately, however, it does this at the expense of detailed information about particular links. Another possible approach to the clutter problem is to omit information about geography.

C. Matrix Display

Like a linkmap, the matrix display concentrates on the links of a network. It tries to address two fundamental problems encountered by the geographic display of network links:

- undue visual prominence may be given to long lines and
- long (e.g., transcontinental) lines may overplot other lines.

As noted before, the linkmap partially addresses this problem by thickening the links in proportion to the statistic values and drawing the important links last.

Both problems may be solved simultaneously, however, by using a matrix display, which de-emphasizes the geography by displaying the network in a matrix form with each matrix element allocated to a link. Each node is assigned to one row and column with the (i, j) and (j, i) matrix elements associated with the j-to-i and i-to-j links. If the link data is not directed, both of these elements are assigned the same value. Fig. 5 demonstrates this technique with each of the small squares corresponding to one of the potential half-lines in Fig. 3, and the colored squares corresponding to the realized lines. The nodes are arranged in approximate geographical order with west-to-east along the horizontal axis and correspondingly along the vertical axis. The matrix representation shows that the earthquake overload is highly focused on five nodes experiencing major incoming blocking and on three others with some incoming blocking. There is one node with outgoing blocking to nearly every other node indicated by the vertical column. This result, obscured in the linkmap, is obvious with the matrix display.

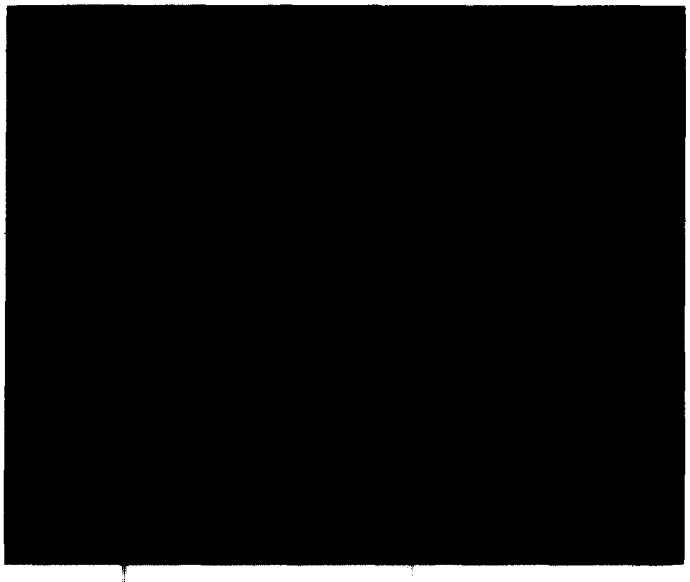

Fig. 5. Network Overload As Matrix. The same overload as in Fig. 3 shown using a matrix representation instead of a network map. The nodes are shown along the rows and columns in approximate west-to-east order in matrix form, with columns corresponding to "from" nodes and rows corresponding to "to" nodes. At the intersection of each row and column there is a square whose color codes the link statistic. The colored squares on the left and bottom correspond to the lines on Fig. 3. The nonsymmetry is due to the directed nature of the traffic.

It is clear the matrix display does well in comparison with the linkmap when there are many lines on the display. What it gives up for this is the information about geography, since the order of the nodes in the rows and columns is only slightly related to their geographic positions. Even if the nodes have no natural geographic layout, the inferences drawn from the matrix display are likely to be influenced by the order in which the nodes are assigned to rows and columns. This ambiguity of row and column order is the chief drawback of the matrix display—a display with one choice of orders may be clear, but the same display with rows and columns permuted may be hard to interpret.

III. Parameter Focusing

The static displays in Figs. 2 through 5 are natural and work well for many network data sets. The maps inherently show the spatial relationships between the nodes. The lines in Figs. 2 and 3 are ideal for showing link-related statistics while maintaining the geographic context. The glyphs in Fig. 4 show the nodal statistics, and the squares in Fig. 5 give each link the same visual prominence. But each of these displays must typically be constructed with great care to avoid visual overload. For example, if the network has too many links, say more than 10% of the $n^2/2$ possible links with n nodes, a linkmap may become cluttered and visually confusing. For a nodemap, as in Fig. 4, the glyphs must be carefully sized so that they overlap as little as possible. For network maps it may be necessary to modify the spatial layout if there are too many nodes too close together, as we did for the AT&T network map. The breakpoints used to assign colors for Fig. 5 must be carefully chosen to ensure that the colors in the squares convey meaning.

Each network display is determined by a group of display parameters as well as by the particular network data. The parameter values control the characteristics of the display, and we call the process of selecting these values *parameter focusing*. The idea is similar to adjusting parameters on a camera, e.g., focus, f-stop, exposure, before snapping a picture. Once the full set of parameters is determined, a static display can be produced because the set of parameters defines a particular map.

Of course, it takes talent, and sometimes luck, to select the proper parameter values. Just as a photographer brackets exposures to make sure that just the right amount of light is used for a photograph, a series of static displays can be made with a range of parameter values.

This tedious process, however, may be made more efficient through *dynamic* parameter adjustment. The analyst manipulates the display parameters dynamically while watching the display change; good parameter focusing is achieved when the display shows meaningful information about the data. This is akin to instant or video photography, and it makes it much easier to get the optimum image.

What parameter values are set in the focusing process? There are several broad classes:

Statistic: The choice of displayed statistic is the most important parameter. This parameter must be easy to select and to vary. For example, it may be informative to move back and forth between absolute overload and percentage overload. Transformations may also be needed: square roots, logarithms, and other computational tools should be readily available.

Levels: This parameter allows the analyst to decide which data to display or, conversely, which data to suppress. Our linkmap tool controls the levels parameter with a two-sided slider that restricts the display to any range of the displayed statistic. A generalization of this technique—brushing [1]—may be applied to the color scale. Our doubled-edged color slider is a special case of a more general slider enabling a user to select arbitrary, disconnected ranges on the scale [2].

Geography/Topology: This parameter allows display of an appropriate subset of the data, based upon network geography or topology. Geographic restriction can be accomplished by a general zoom operation that allows any rectangular subregion to be displayed. Control of network topology enables the analyst to deactivate (or reactivate) any nodes and associated links in the map. The operation is quick, making it possible to eliminate node pairs involving any particular location or group of locations, and thereby to concentrate on the other parts of the map. This operation also allows the analyst to focus on a particular location or set of locations by deactivating all locations and selectively reactivating the location or locations of interest.

Time: When the data at each node or link is varying through time, it is important to select the proper time point to display. In some cases, the proper time is known a priori; in others, it must be determined from a mass of available data. Our displays accommodate spatial time-series data, allowing the analyst to vary the displayed time point at will, and even automatically, providing snapshots at particular times or producing an animation. This enables the analyst to focus on the most interesting periods or to look for changes in network flows. Often, it is desirable to identify an interesting time period using one display and to investigate it in more detail with a different display.

Aggregation: This parameter is related to the displayed statistic parameter. It is important that the analyst have tools to aggregate statistics over geographical regions or logical subsets of a network. Although this is currently not a dynamic feature in our software, it is easy to use the underlying data management software to aggregate statistics over related locations or geographical regions.

Size: The size parameter controls the overall size of the symbols drawn on the map. Adjusting the size of the symbols allows the analyst to produce a map with symbols large enough to convey information yet small enough to avoid excessive interference with other symbols. This is a difficult problem to solve computationally, yet is simple when done interactively—human perception is a powerful tool for setting the size parameter appropriately. Our link map uses *line shortening* and *line width varying*, and our node map uses *symbol sizing*. With line shortening, the line connecting two locations is drawn only part-way from one location to the other. This helps to address the problem that lines running across the map can obscure what is happening in the middle of the map. The line width redundantly encodes the statistic along with color. Making the important lines thicker helps them stand out and be visually more prominent. With symbol sizing, the size of the rectangle around each location is adjustable. This prevents large rectangles from obscuring neighboring locations.

Color: We use color to encode the statistic values on our displays. By carefully choosing colors we can highlight the important data. On our linkmap and matrix displays the color is determined by the color slider. Adjusting the colors is a dynamic operation. The user may interactively move the boundaries between the colors and add or subtract individual colors. With many types of network data the distribution is skewed and so choosing colors naively is ineffective and will result in nearly all data items being represented by the same color. In nodemap we use color to encode differences in aspect ratios for rectangles and as a threshold for circles. The tall, thin rectangles are green, the short, fat rectangles are red, and the approximately square rectangles are white. Using a slider, the analyst controls how extreme the aspect ratio must be in order for the rectangle to be colored. This way color can be used to highlight important differences. Similarly, for circles the slider controls a threshold that determines how large the circle must be to be colored red.

We illustrate these techniques by continuing our analysis of the AT&T network data. Fig. 6 shows the same data as Fig. 3, except the half-lines are shortened. Fig. 6 also shows the sliders and other controls that allow dynamic parameter focusing. Shortening the lines reduces the visual clutter and neatly solves the line overplotting problem. In Fig. 3 the mass of transcontinental lines completely obscures the middle of the country, making it impossible to see network activity in the Midwest. In Fig. 6 the shortened line segments point east-west, showing that all of the network overflow was into and out of the West Coast. If there had been an overload in the middle of the country, then some of the nodes would have had circular fan-outs like the West Coast nodes.

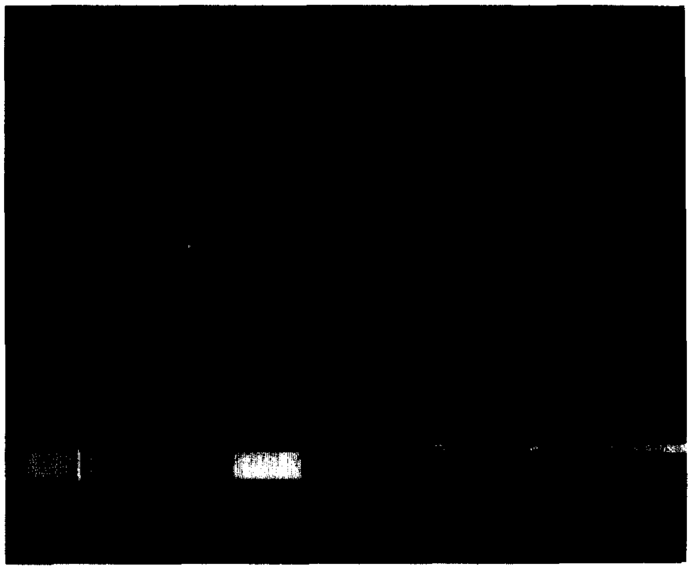

Fig. 6. Line Shortening. The figure shows the same data as Fig. 3, except the half-lines are drawn only part way between the nodes that they connect.

Network map displays often show exception statistics, like overload, that involve a small fraction of the links because otherwise they become overwhelmed by information (even on Fig. 3 less than 10% of the links are displayed). An interesting pervasive statistic (one involving most of the links between nodes) is the percentage of idle capacity of each link. Fig. 7 shows the percentage of idle capacity for links into and out of one node near Chicago. We use geographic restriction to focus on a particular location, thereby enabling us to display a pervasive statistic without the network map becoming too cluttered. To create Fig. 7 we have turned off all of the nodes and then momentarily turned on one of the nodes by touching it with the mouse.

There is idle capacity between Chicago and every other city, but the interesting aspect of this display involves the amount of idle capacity. At this time period, early evening in Chicago after the businesses have closed and before the evening residential traffic picks up, the thick colored lines indicate high percentages of idle capacity to nearly every other city except for the Bay Area and Southern California. Even to these areas where there was heavy blocking during this time period, there is a small amount of idle capacity. The reason for this involved network management controls. AT&T network managers directionalized the calling traffic, giving priority to traffic out of the disaster area while restricting the inbound traffic. The small amount of idle capacity ensured that calls out of the Bay Area could complete, but was not available for calls into the Bay Area.

Fig. 7. Idle Network Capacity. The percentage of idle capacity on links into and out of a Chicago node. By turning off all nodes and interactively turning on selected nodes, we can study a pervasive network statistic.

There are several issues with parameter focusing. First, the space of all possible parameter values is large. Second, most combinations of the parameters do not lead to understandable displays. Third, the displays are sensitive to particular values of the parameters. To address these issues we have developed a network visualization system called SeeNet that provides interactive control over display parameters. The figures in this paper are SeeNet screen dumps.

In the next section we describe the ways in which the data analyst interacts with the displays described earlier, using SeeNet.

IV. Direct Manipulation

With parameter focusing there can be many different combinations of the parameters, most of which do not lead to interesting and interpretable displays. A solution to the problem of selecting the interesting parameter values involves direct manipulation [3]. SeeNet allows the analyst to modify the focusing parameters while continuously providing visual feedback, enabling the adjustment of the parameters to produce informative displays. The idea is, as with adjusting the focus on a camera, that the display should update smoothly, showing the results as the parameter controls are adjusted. For this to be effective, the computer response must be nearly instantaneous (less than 50 milliseconds).[2] When a parameter is being adjusted, every mouse movement causes a parameter to be modified. Manipulating a slider might involve hundreds of small, incremental mouse movements. Point and click user interfaces are less effective for parameter focusing because each adjustment requires a click. The effects of mouse movements are instantaneously reflected in the display, providing user feedback and determining whether or not the adjustment is making the display more informative. Using direct manipulation, the analyst may search a rich parameter space to focus on interesting displays.

A. Identification

SeeNet allows the interactive identification of nodes and links. The analyst identifies a node in SeeNet by touching the node with the mouse (no need to press a button), which causes the node to become active and turn a different color. In linkmap, identifying a node additionally causes its name to appear on the display; in nodemap, identifying a node shows its name and the data values for that node. This ability is crucial for an effective interactive display, where much information has been suppressed to make the display, making it possible to retrieve the suppressed information quickly.

A more complex form of identification is available in linkmap. Since two nodes determine a particular link, the analyst can first indicate an *anchor* node with a button press. Later node identifications then include information about the link between the identified node and the anchor node. In Fig. 6 the anchor node is `SKTNCA` and the current node is `OKLDCA`. The direct link statistics are displayed in the right-hand corner of the screen, `170` from `OKLDCA` to `SKTNCA` and `5827` in reverse. A time-series plot for the anchor-to-current link is displayed within the time slider.

Identification is also possible in the matrix display. In that case, identifying a link is easy, as the analyst can just point to the corresponding cell.

B. Linkmap Parameter Controls

There are three vertical sliders in linkmap to manipulate the line length between the links, the line thickness, and the animation speed (see Fig. 6). To obtain the pleasing result in Fig. 6 from Fig. 3, we manipulated the line length slider, moving it from `Long` toward `Short`. The line width slider sets the thickness of the maximum link and at the `Thin` end makes every line one pixel wide.

There are two horizontal controls, an interactive color legend and a time slider. The color legend functions as a double-edged slider with upper and lower thresholds. By manipulating the thresholds, the analyst may reduce the visual complexity of the display by eliminating those lines whose statistic values are not between the thresholds. A toggle determines whether the suppressed lines are inside (between) or outside the threshold bars. The double-edged slider enables an analyst to threshold from two directions, or using the toggle, to show simultaneously the lines with both high and low statistic values, while

[2] This corresponds to 20 frames per second and, in our and our users' experience, interactive network displays with significantly slower response time felt sluggish and were noticeably less satisfying.

ignoring those between. The time slider sets the current period and the area inside the slider is used to display a time-series for the active node or link.

C. Matrix Display Parameter Controls

The matrix display also uses linkmap's interactive color legend and time slider parameter controls. In addition, it has the capability to permute the rows and columns. As mentioned earlier, the orders of the rows and columns are an important design problem for the matrix layout. Careless ordering may make the display nearly impossible to interpret. Bertin [4] constructed mechanical systems for experimenting with different row and column orders and suggests a strategy for permuting the orders. In SeeNet, the analyst repositions a row or column using a *drag-and-drop* action. Depressing the left mouse button while touching a row or column label grabs it, and moving the mouse then drags it to a new position, where it is dropped into place when the mouse button is released.

D. Nodemap Parameter Controls

Three vertical sliders in nodemap manipulate the symbol size, animation speed, and color sensitivity level. Choosing an effective symbol size is a straightforward and intuitive operation for the person manipulating the slider.

There are two frequently used glyphs: a circle to code a single statistic and a rectangle to code two statistics (width and height). Other glyphs [5], encoding three or more variables, are possible, though we have had less success with them. By default the glyphs are white. Color is used to highlight the unusual glyphs: for circles the largest circles are colored red, and for rectangles color is tied to the aspect ratio. Short, fat rectangles are colored green, while tall, thin rectangles are colored red. The color sensitivity slider controls the cutoff values for color changes. We encode negative values using dashed instead of solid lines.

E. Animation

Animation is a useful technique to scan data from many time periods and is crucial for analyzing time-varying data from large networks with many time periods. If the data from each time period is correlated, the resulting animations show smooth evolutions. Unusual changes are then jarring and readily apparent. SeeNet's animation capability causes the computer to walk continuously over all of the time periods, with the `Fast-Slow` slider controlling the animation speed. Manual animation is possible by dragging the time bar forward or backward with the mouse, with the display updating continuously.

F. Zooming and Bird's-Eye

In SeeNet an analyst may zoom into any rectangular subregion by depressing the `Zoom` button and selecting a region. The selection action is accomplished by clicking on the center of the region and sweeping out. Center-to-edge sweeping differs from the traditional mechanism for sweeping out windows on workstations, which is to select a corner and sweep out to the other corner. We found this action difficult to use because it was hard to center the windows correctly when sweeping corner-to-corner. Our technique, center-to-edge, assures that the zoomed area will be centered appropriately and sized to cover the area of interest. Our zoom operation preserves the map aspect ratios. For the traffic matrix, the zooming operation is similar; selecting a rectangular subregion causes this region to expand to occupy the whole display.

A general problem with zooming in arbitrary abstract networks is maintaining global context—it is easy to get lost. SeeNet's bird's-eye view in the upper left-hand corner helps maintain the global context by showing the zoomed area on the original display. It is also possible to manipulate the zoomed area (in particular, to pan it) using the bird's-eye view.

We have also experimented with context-specific zooming to focus in on a particular region or state and also with continuous zooming. In our implementation of continuous, variable-gain zooming, the user selects the center of a sub-region and moves the mouse away to control the magnification. As with a variable zoom lens on an expensive camera, the zoomed region changes smoothly, tracking the mouse movements.

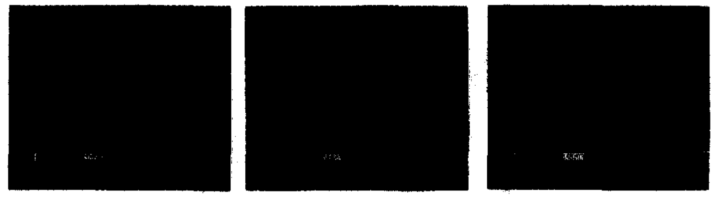

Fig. 8. Interaction Between Links And Zooming. The zoomed area is in the interior of the network shown in Fig. 3. The left pane shows all lines, the middle pane shows all lines terminating within the zoomed area, and the right pane shows all lines that both originate and terminate in the zoomed area.

In linkmap there is an interaction between the lines connecting the links and the zoomed view. There may be lines that: 1) pass through the zoomed area, but with both endpoints outside the zoomed area, 2) terminate on one node in the zoomed area, or 3) are completely inside of the zoomed area. The best choice for resolving this interaction depends on the application and the particular question of interest. Fig. 8 shows a zoomed view in the middle of the network from Fig. 3. The panes from left-to-right show the different interactions between the zooming and which lines are displayed. In the left pane, all line segments intersecting the display are drawn. Clearly, it is too busy to interpret. The middle pane contains any line segment with at least one endpoint in the display (that is, it excludes any segments both of whose endpoints lie outside the display). In the right pane, only lines that both begin and end inside the display are shown (in this case, none!). The middle and right panes show that there are many overloads between nodes inside the display and nodes outside, but none between nodes within the display. There are buttons on the display that determine which of these views to display, and the analyst is able to toggle quickly between them.

G. Conditioning

In many situations there are several related statistics for each link such as capacity, current utilization, and predicted utilization. Our approach to understanding multivariate link data involves conditioning. An analyst selects an interesting range for one or more background variables and sets the display to show a foreground variable. The conditioning operation filters out all links whose background variables are not within the selected ranges, visually showing the intersection between the sets.

Conditioning is a key technique for answering questions involving *and*. An important question, for example, might involve the overload on the larger links. The overload on the small links is less important because these links do not carry much traffic and is therefore less interesting. To show the large-link overload the analyst would use the double-edged slider to filter the smaller capacity links and set the display to show the overload. Conditioning and suppressing the links with smaller capacities visually focus the display on the large-link overloads.

H. Sound

In visualization systems, sound is another independent channel to the user for conveying information [6]. SeeNet uses sound in three ways:

- **Node State Changes**: When the analyst selects any node and either activates or deactivates it, SeeNet confirms the operation with an audible bell. There are different bells for activation and deactivation, and we find, in practice, the audible response to be an effective confirmation of the selection.
- **Conveying Slider Values**: As the sliders are manipulated, SeeNet plays a tone of varying pitch that tracks the slider bar's position. The pitch increases as the bar is moved up and decreases as the bar is moved down.
- **Animation Frame Changes**: During animations, SeeNet produces a "click" every time the current time period changes and a "bell-ringing" sound whenever the animation repeats. In practice, this is effective because the sound frees the viewer's attention from the time period slider. The audible clicks sound like a metronome keeping time, and the bell indicates that the animation is restarting. During our analysis of one particular dataset we discovered an error when, during the animation, we saw that the display changed only every three clicks. Data for each time period was erroneously repeated three times, but we had not noticed this until we simultaneously heard the clicks and watched the display.

V. Further Examples

The tools we have described are generic, and we have applied them to a variety of situations, some of which we wish to discuss in this section. We will use two examples: the CICNet packet-switched data network and an e-mail communications network.

A. CICNet

The CICNet network is one of the regional data networks of the U.S. portion of the Internet. It connects 13 universities and research facilities in the Great Lakes region. Fig. 9 shows large circles at each of the facilities. These large circles represent the packet routers. Lines are drawn between the routers to show which pairs of them are physically connected. The smaller circles represent local area networks attached to the routers. Each router counts the number of bytes it sends to and receives from each of the local area networks and the other routers to which it is connected. The nodes of Fig. 9 occur where line segments terminate at routers, and at each node we have drawn a rectangle, just as in Fig. 3. The size of the rectangle is determined by the number of incoming and outgoing bytes at that point over a 10-minute period.

Notice that unlike the AT&T example, the underlying map here is schematic, not geographic (though the layout of the routers is vaguely similar to the real geographic layout). The underlying map is actually a parameter to our nodemap tool, so that no change to the tool is needed to analyze this quite different situation.

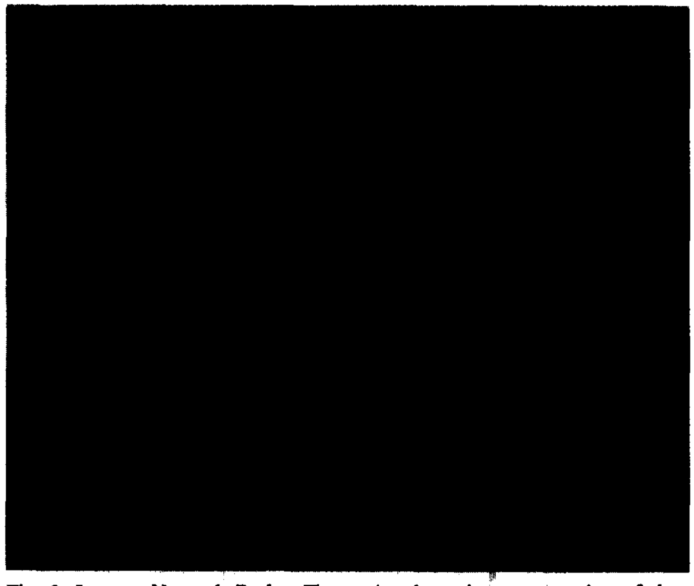

Fig. 9. Internet Network Packet Flows. A schematic representation of the CICNet regional network, showing routers (larger circles), local area networks (smaller circles), connections between them (line segments), and byte flows along the connections (rectangles where the line segments terminate on the router circles). A flow from the ARGON (Argonne National Labs) to the MINN (University of Minnesota) local area networks can clearly be discerned, in spite of the lack of explicit information about this pair.

Another difference between this example and the AT&T example is that the node statistics are not derived from an accumulation of link statistics. That is, thinking of the local area networks as the sources and sinks of the bytes flying around the network, we do not have direct information about the total traffic between arbitrary pairs of these sources and sinks. On the other hand, as Fig. 9 spectacularly demonstrates, it is immediately possible to infer such information—notice that there is a strong flow from Argonne to University of Minnesota in the 10-minute period shown here, as indicated by the sequence of red and green rectangles. Other, smaller flows are discernible as well.

Fig. 9 is one frame from a series of such frames representing all 10-minute periods for one month. Animating these, as described earlier, gives an excellent dynamic view of the network, and has led to some interesting insights [7].

B. E-mail Communications

Within the technical community, electronic mail is widely used for interpersonal communications. [8] In our location, members of the technical staff receive 30 to 40 messages per day. For nearly a year we logged the sender, receiver, message size, and time for every e-mail message sent or received by members of one of our departments at AT&T Bell Laboratories who volunteered for this study. This dataset is interesting because there is no natural geography or spatial layout associated with the network and there are no predetermined links.

In Fig. 10, each sender and receiver is represented as a node with the link statistics encoding the number of e-mail messages sent between the individuals in a time period (one month). During this period there were over 1500 different user IDs exchanging e-mail, and we have restricted the display using the slider to show only those links corresponding to 20 or more messages. We have aggregated the messages so that the link statistics are nondirectional.

Positioning nodes and links for drawing arbitrary graphs is a delicate operation involving subtle trade-offs [9]. There is a fundamental perceptual issue involving the node placement and the length of the links connecting them. Nodes that are close to each other are perceived to be related, and yet the link between them does not receive much visual real estate, making it hard to see. Conversely, distant nodes are not perceived to be related and yet are connected by a long, visually apparent link. Some of the best graph-drawing algorithms use this perceptual tension for conveying information related to particular applications.

Fig. 10. Department E-mail Communication Patterns. Each node corresponds to a user, and the links encode the number of electronic mail messages sent between the users.

The node positioning in Fig. 10 uses a spring-tension model solved by synthetic annealing so that users exchanging large quantities of e-mail are close to each other. This results in a visually interesting and easily interpreted display [10]. User ID `hastings` is in the "e-mail center" of the department. `Hastings` is our resident computer expert, system administrator, and lead guru, and he communicates heavily with everyone else via e-mail. Other communications foci involve `eick`, `cope`, `jcr`, lead investigators, and `dorene`, our department secretary. Around the edge of the display are newer employees, members of other departments, and even automated e-mail response systems such as `TheDailyQuote`.

C. World Internet

In an example that covers a wider geographic region, we have used SeeNet to display statistics from the Internet. Fig. 11 shows country-to-country traffic across the NSFNET/ANSnet[3] backbone for the week of February 1-7, 1993. The colors and widths plotted are proportional to square root of the packet counts (square roots are often an appropriate transformation for counted data). To make the picture comprehensible, clutter was reduced by adjusting the lower threshold to display only those country-to-country links that carried more than one million packets that week. The data was displayed on a world map produced by S [11]. From the figure, the primary connections from the United States to other countries are visible (to many countries in Europe, Canada, the Far East, and Australia) as well as certain transit routes, for example from Australia to Europe. By deactivating the node for the United States, it is possible to focus on the transit traffic; similarly, with all the nodes deactivated, a mouse touch reactivates the current node and shows the packet counts to and from that country.

VI. The SeeNet System and Its Users

Originally we developed the SeeNet network visualization system for our research in network data analysis. Our objective and motivating example was to develop a system able to cope with the volumes of data generated by the AT&T long distance network traffic. Our static displays were modeled on existing network displays, and we invented the interactive techniques to overcome the display clutter problems associated with displaying large networks. SeeNet and the techniques embodied in it have evolved as the result of experience over the last several years in using it to visualize complex network data.

SeeNet has been used by many groups within AT&T to analyze network traffic, study overloads, engineer private line capacity changes, understand facility churn, display packet network traffic, and visualize network simulation output. It has also been embedded in the *PATTERNS* Network Operations Support System and is used daily by engineers for monitoring the performance of the FTS2000 Federal Telecommunications network.[4]

The original version of the SeeNet system ran on a Silicon Graphics IRIS workstation, although it has subsequently been ported to most X11-based Unix workstations, and there is also a limited version running under MS Windows. The current

[3] Thanks to Hans-Werner Braun at the San Diego Supercomputer Center and the NSFNET partnership for making the data available. It is in /pub/scsc/anr/data, available by anonymous ftp from ftp.sdsc.edu.

[4] This government network is currently the largest and most advanced private custom network in the world with well over a billion minutes of annual calling generated by over one million users at thousands of different locations.

version of SeeNet is written in C++ [12] using the Vz Visualization library. [13] Vz is an object-oriented, cross-platform library (X11, OpenGL, and MS Windows) that embodies linked views, direct manipulation, and data abstractions in a selective manner for building novel, production-quality visualization systems. In total SeeNet is about 6000 lines of code, divided into five views: the network map, the time slider, the interactive color scale, the bird's-eye, and the control panel. Each view is implemented as a Vz C++ class, is independent of the other views, and uses a publish and subscribe mechanism for coordinating responses to user actions.

Fig. 11. Worldwide Internet Traffic. Traffic on the Internet, square root of packets transmitted from country to country across the NSFNET backbone during the first week of February 1993.

For a network with 100 nodes, for example, there are 9900 directioned node pairs and, for adequate performance with a pervasive network statistic, it is necessary to draw that many colored line segments several times per second. For SeeNet's parameter focusing to be most effective, the computer must respond continuously to control changes. To achieve sufficient performance on slower workstations and, in particular, on MS Windows personal computers, SeeNet uses several programming tricks. Each of the views is double buffered, either using hardware support if it is available or by rendering to an off-screen image. The node positions are stored in a quad-tree so that the identification operations are fast. To avoid redrawing the main display, the map is drawn in an underlay bitplane, and highlighting operations occur in an overlay bitplane so that they do not damage the main display. For systems without hardware overlay and underlay support, we simulate them using color map manipulation. At initialization time we precalculate as much as possible to avoid run-time overhead. For example, we sort the links in ascending order so the links are automatically drawn in the appropriate order.

The background maps, link statistics, node statistics, and colors are all input parameters to the program, making it easy to work with a variety of underlying maps and to explore different statistics. In early versions of SeeNet we used the S language [14] for data analysis. S provides data management, static graphics, computational analysis, and transformations in support of the dynamic operations. Using S, it is easy to investigate data transformations. Traffic flow statistics, for example, tend to be highly skewed with a few extreme values, so a logarithmic or square-root transformation can be used to ensure a more uniform distribution of the statistic. Map displays may have only a handful of red lines (the highest interval) out of thousands of links. A logarithmic or square-root transformation can be used to ensure a more equal distribution of colors. Other computations can be carried out prior to display. For example, given data for two adjacent time periods we might want to compute the change in the statistic from one period to the next. The resulting signed values can be displayed on the map to show where the flow is increasing or decreasing. Here the analyst would probably select a nonstandard color scale such as red for increasing, blue for decreasing. In addition, S can be programmed to provide a menu-driven user interface to select which data should be displayed in SeeNet.

VII. Discussion

In this section we briefly review some of the other work that has been done in this area, and then we compare and contrast our different techniques for displaying network data.

A. Related Research

Because of the importance of network data, there is a rich history in network visualization. Some fundamental early work is due to Bertin [4] who uses both node and link representations as well as matrix representations. Bertin also introduces the idea of interactively manipulating a network display using specially built, manual tools.

The data analysis setting we have described in this paper involves both a network and data on that network and concentrates on how to analyze the data. If we leave aside the data, we find much that has been written about displaying and attempting to interpret the structure of networks (or graphs, as they are usually called). For example, Fairchild, Poltrock, and Furnas [15] describe the SemNet system for displaying and manipulating a three-dimensional view of a possibly quite large network. One of the problems is how to create such a view, given the abstract structure of the network. Our earlier CICNet example presented a layout that we constructed by hand, but for a large network this is clearly impractical. This is a general problem—graph layout—about which much has been written in recent years, including several simple techniques in the paper under discussion.

Again focusing on the network rather than the network data, Sarkar and Brown [16] describe an innovative fisheye distortion for visualizing the structure of sparse networks. Their system, like ours, is dynamic and supports smooth transitions between the views. Their displays are engaging and interactive but do not address the fundamental problem in network data visualization: coping with the display clutter caused by too many link crossings.

Among other things, Paulisch [17] discusses both the problems of network layout and ways to focus on particular parts of the layout. One interesting idea involves what she calls "edge concentration," in which a local group of highly connected nodes is simplified by introducing a new pseudo-node with edges to each of the original nodes in the local group. We have thought of using this idea, but in a slightly different way, by collapsing such a group of highly connected nodes into a single node and showing only the collapsed node, with aggregated links to all nodes connected to any of the original nodes.

However, what is of more interest in comparing their work with ours is the means by which they have attempted to visually focus the large amount of displayed information: showing only a prenamed subset of the network, restricting the three-dimensional viewing angle, and using fisheye views. The last is a novel method of what could be called "gradual focusing," in which the piece of the network close to the focus of interest is displayed in complete detail, while only the more important features are shown for parts of the network further away. This technique can be introduced gradually, so that the level of detail varies with the distance from the point of interest.

Unwin, Sloan, and Wills [18] describe an exploratory analysis of oil imports and exports to and from European countries. The nodes, representing countries, are represented by symbols positioned on a map of the world in the center of their countries, and the links encode the oil imports and exports between the countries. This problem is simplified because there are not many European countries and, in particular, only a handful of oil-exporting countries. The authors use interactive methods, such as thresholding, for analyzing their oil data.

Eick and Wills [10] use aggregation, hierarchical information, node positioning, and linked displays for investigating large networks with hierarchies. They focus on abstract hierarchical networks with no natural spatial layouts. They represent networks using node and link diagrams, with shape, color, and other visual characteristics coding nodal information and with color and line thickness coding link information. They use node positioning and hierarchical aggregation to address the map clutter often present in large complex networks.

Researchers at NCSA [19] have added 3D graphics to their network maps. They display animations of Internet packet traffic with the network backbone raised above the network map. The links are color coded according to the size of the packet flows. Their techniques result in visually interesting displays of aggregate network traffic, especially with their creative use of 3D to overlay additional information.

Casner [20] describes an automated system for displaying airline traffic. His system is goal directed; it uses different graphical layouts depending on the particular question currently of interest. Although theoretically interesting, the system is not practical for large networks.

In other related work focusing on database visualization Consens et al. have developed a sophisticated interactive graph visualization system [21] for visualizing hygraphs. Hygraphs are a generalization of standard graphs containing nodes, links, labels, and container information. They have developed a sophisticated system called Hy^+ for browsing of hygraphs and applied it to visualize software structure [22] and network management data. [23] One of the most innovative aspects of their system is its foundation, a powerful programming language for manipulating hygraphs. Their focus is on extracting information from a complicated hygraph using interactive queries and displaying the results using their powerful browser.

Koike [24] describes a system called VOGUE, designed to display communication patterns in parallel processing computer systems. It is generally concerned with displaying networks, with nodes and links positioned in 3-space and rendered with various symbols, sizes, and colors. The system allows interactive selection of viewpoints. The author argues that 3D representations of data allow the user to "see two relations simultaneously and to focus on each single relation without changing mental models."

There are many possibilities for using sound to enhance data analysis, though we feel that efforts to date have not been notably successful. To see what others have tried, consider, for instance, Bly's work [25] or the paper by Mezrich, Frysinger and Slivjanovski [26]. More recently, Gaver investigated the use of sound in user interfaces [27], [28].

SeeNet's interactive user interface and parameter focusing techniques are motivated by statistical research in dynamic graphics [29], [30]. The essential idea involves the interactive controls for focusing the display. One special case of focusing, suppression, eliminates display clutter by interactively restricting what appears on the display. This technique is most efficient when it is dynamic as with our double-edged color slider [31], [32].

Recently, the Human-Computer Interface community has discovered the usefulness of interactive filtering techniques for building effective visualizations. Shneiderman describes three systems for visualizing real estate, the periodic table of elements, and movies that represent data spatially and uses double-edged sliders. [33] Ahlberg and Shneiderman also recognize that a nearly instantaneous response is critical for the effectiveness of the technique. [34]

B. Link Data

For link data, particularly for networks with geographic information, the spatial layout is visually more easily interpreted than the matrix representation. For small numbers of links, presenting the information on a map results in a more meaningful display because it preserves the network context. Maps present the spatial information clearly, and it is natural to layer the link information on top of the spatial layout.

The map representation breaks down when the number of links is large, because the display becomes too cluttered. This is particularly a problem when the links extend across a significant fraction of the display, as do the transcontinental lines in Fig. 3. This occurs, for example, when a pervasive statistic is displayed. There are three possible solutions to resolve the map-clutter problem:

- Reposition the nodes to minimize the number of long links (see Fig. 11 and Eick and Wills [10]). This can be done interactively.

- Apply interactive parameter manipulations such as thresholding and filtering to reduce the visual complexity.
- Use the matrix representation.

The matrix representation neatly solves the link overplotting problem in this example, because the links are represented by squares, tiled on the display. Since each square is assigned the same amount of visual real estate, the long links are not visually dominant. The resulting display is uncluttered since the squares cannot overplot each other, but loses the easy interpretation and context provided by the spatial map layout.

For ordering the rows and columns in matrix displays we have used geographic position, hierarchical clustering, and direct manipulation. None of these approaches is completely satisfactory, suggesting that more research in this area is needed. Some other approaches can be found in the paper by Friedman and Rafsky [35].

C. Node Data

The nodemap represents node data using variable-sized glyphs spatially positioned on the display. The size slider nicely solves the glyph overplotting problem by enabling the interactive selection of a size large enough to be visual, yet small enough not to cause excessive overlapping of the glyphs. Color is an effective visual cue, drawing attention to unusual glyphs, with the color sensitivity slider preventing overuse of the cue.

We have experimented with different types of glyphs for coding multiple statistics. For example, we tried crosses for two statistics and thermometers for three statistics. For a single statistic we prefer circles, for two statistics we prefer rectangles. For three or more statistics, more complex glyphs are possible, but we have had limited success. In the perception literature [36] researchers have found that humans visually decode area as a fractional power of the statistic that it encodes. This argues that rectangles may be a poor glyph choice. A possible solution to the visual glyph decoding problem is to transform the data, e.g., take square roots, so that area of the glyph is proportional to a meaningful statistic.

For our applications there are three reasons why we find rectangles so appealing. The first reason is that they overplot nicely. The second reason involves the aspect ratio. Rectangles with similar aspect ratios encode similar data patterns. By color coding according to the aspect ratio, red for the fat short ones, green for the tall thin ones, and white for the square ones, it is easy to identify the similarly shaped rectangles and therefore related nodes using our pre-attentive vision. [37] The color coding is made even more effective by linking the threshold to a slider. By manipulating the slider, the threshold can be set so that only rectangles with significantly different aspect ratios are colored, thereby focusing our attention on this important set. Finally, we have found that it is easy to compare the horizontal and vertical extents of similarly shaped rectangles by mentally juxtaposing them. This enables us to deduce quantitative results by observing that a designated rectangle may be twice or three times as large as another, for example.

VIII. Summary and Conclusions

This paper describes techniques for visualizing the data associated with large networks. Three static displays provide complementary views:

- a spatial layout with links between the nodes,
- a matrix layout with the links represented by cells, and
- a spatial layout with the nodes represented by glyphs.

For large networks displays produced using these well-known static techniques fail because the displays become cluttered and visually confusing.

To solve the display clutter problem, we invented a suite of parametric techniques embodied in a dynamic graphics software system called SeeNet that enable a user to focus the display and thereby reveal patterns in the network data. The dynamic parameters control:

- the data-based statistic that is displayed,
- which data values should be displayed (e.g., to omit low values),
- geographical or topographical regions that are displayed,
- which time interval is shown for time-based data,
- the level of aggregation of the statistic,
- the overall size of symbols.

Once these parameters are chosen, they completely determine a particular display. However, it is often difficult to choose these parameters numerically. To allow precise control over these parameters, SeeNet uses a direct-manipulation user interface containing sliders, buttons, and mouse-sensitive screen regions allowing the user to manipulate the parameters interactively. As the user modifies the controls, the network display smoothly updates. This parsimonious set of focusing parameters has been delicately chosen based on our experience visualizing a wide range of networks.

We have analyzed many different types of network data using SeeNet, and its features have evolved to meet our data analysis needs. SeeNet incorporates a small, carefully selected set of parameters that accommodate many kinds of network data.

Acknowledgments

We would like to acknowledge Larry Bernstein, Brian Johnson, Ron Kaufman, Eileen Miller, Carol Mummert, Eric Oberer, and Rich Wolf for ideas and encouragement during the formative stages of SeeNet and for careful proofreading of early drafts. Ken Cox implemented the Vz version of SeeNet.

References

[1] Richard A. Becker and William S. Cleveland, "Brushing Scatterplots," *Technometrics*, vol. 29, pp. 127-142, 1987.

[2] Stephen G. Eick, "Data Visualization Sliders," *UIST 94 Conference Proceedings*, Monterey, Calif., pp. 119-120, Nov. 1994.

[3] Ben Shneiderman, "Direct Manipulation: A Step Beyond Programming Languages," *IEEE Computer*, vol. 16, no. 8, pp. 57-68, 1983.

[4] Jacques Bertin, *Graphics and Graphic Information-Processing*, Berlin: Walter de Gruyter Co., 1981.

[5] Jeff Beddow, "Shape Coding of Multidimensional Data on a Micro-

computer Display," *Visualization 90 Conference Proceedings*, Los Alamitos, Calif., pp. 238-246, 1990.

[6] Mark Brown and John Hershberger, 'Color and Sound in Algorithm Animation," *IEEE Computer*, vol. 25, no. 12, pp. 52-63, 1991.

[7] Richard A. Becker and Allan R. Wilks, "Final Report: Network Traffic Analysis Project," *BBN Subcontract agreement #20510*, Feb. 1993.

[8] Michael F. Schwartz and David C.M. Wood, "Discovering Shared Interests Using Graph Analysis," *Communications of the ACM*, vol. 36, no. 8, pp. 78-89, 1993.

[9] Emden R. Gansner, Eleftherios E. Koutsofios, and Stephen C. North, "A Technique for Drawing Directed Graphs," *IEEE Trans. on Software Engineering*, vol. 19, no. 3, pp. 214-230, 1992.

[10] Stephen G. Eick and Graham J. Wills, "Navigating Large Networks with Hierarchies," *Visualization 93 Conference Proceedings*, San Jose, Calif., pp. 204-210, Oct. 1993.

[11] Richard A. Becker and Allan R. Wilks, "Maps in S," *AT&T Bell Laboratories Statistics Technical Report*, anonymous ftp at research.att.com:stat/doc/93-2, 1993.

[12] Bjarne Stroustrup, *The C++ Programming Language*, Reading, Mass.: Addison-Wesley, 1987.

[13] Visualization Group, *Vz Visualization Library: User's Guide Version 1.0*, AT&T Bell Laboratories, Jan. 1995.

[14] Richard A. Becker, John M. Chambers, and Allan R. Wilks, *The New S Language*, Pacific Grove, Calif.: Wadsworth & Brooks/Cole, 1988.

[15] Kim M. Fairchild, Steven E. Poltrock, George W. Furnas, "Three-dimensional Graphic Representations of Large Knowledge Bases," *Cognitive Science and Its Applications for Human Computer Interaction*, pp. 201-233, 1988.

[16] Manojit and Sarkar, and Marc H. Brown "Graphical Fisheye Views of Graphs," *Communications of the ACM*, vol. 37, no. 12, pp. 73-84, Dec. 1994.

[17] Frances Newbery Paulisch, *The Design of an Extendible Graph Editor*, Dissertation for the Department of Computer Science, University of Karlsruhe, Jan. 1992.

[18] Anthony R. Unwin, B. Sloan and Graham J. Wills, "Interactive Graphical Methods for Trade Flows," *Proceedings of New Techniques and Technologies for Statistics*, to appear.

[19] National Center for Supercomputing Applications, "A Visualization Study of Networking," video tape, Donna Cox and Robert Patterson, Co-Principal Investigators, University of Illinois, 1991.

[20] Stephen M. Casner, "A Task-Analytic Approach to the Automated Design of Graphics Presentations," *ACM Trans. on Graphics*, vol. 10, no. 2, pp. 111-151, 1991.

[21] M.P. Consens, F.Ch. Eigler, M.Z. Hasan, A.O. Mendelzon, E.G. Noik, A.G. Ryman, and D. Vista, "Architecture and Applications of the Hy^+ Visualization System," *IBM Systems J.*, vol. 33, no. 3, pp. 458-476, 1994.

[22] M.P. Consens, A.O. Mendelzon, and A.G. Ryman, "Visualizing and Querying Software Structures," *Proceedings 14th Int'l Conf. on Software Engin.*, pp. 138-156, 1992.

[23] M.P. Consens and M.Z. Hasan, "Supporting Network Management through Declaratively Specified Data Visualizations," *Proc. IEEE/IFIP Third Int'l Symp. on Integrated Network Management, III*, pp. 725-738, April 1993.

[24] Hideik Koike, "The Role of Another Spatial Dimension in Software Visualization," *ACM Trans. on Information Systems*, vol. 11, no. 3, pp. 266-286, 1993.

[25] Sara Bly, "Sound and Computer Information Presentation," *Lawrence Livermore Laboratory Technical Report UCRL-53282*, March 1982.

[26] J.J. Mezrich, S. Frysinger, and R. Slivjanovski, "Dynamic Representation of Multivariate Time Series Data," *J. of the Am. Statistical Assoc.*, vol. 79, pp. 34-40, 1984.

[27] William Gaver, "The SonicFinder: An Interface that Uses Auditory Icons," *Human-Computer Interaction*, vol. 4, pp. 67-94, 1989.

[28] William Gaver, "Synthesizing Auditory Icons," *CHI 93 Conf. Proceedings*, pp. 228-235, 1993.

[29] Richard A. Becker, William S. Cleveland, and Allan R. Wilks, "Dynamic Graphics For Data Analysis," *Statistical Science*, vol. 2, pp 355-395, 1987.

[30] Willian S. Cleveland and Marylyn E. McGill (eds.), *Dynamic Graphics for Statistics*, Pacific Grove, Calif.: Wadsworth & Brooks/Cole, 1988.

[31] Richard A. Becker, Stephen G. Eick, Eileen O. Miller, and Allan R. Wilks, "Dynamic Graphical Analysis of Maps—31 minute videotape," *Bell Laboratories ITV Facility*, Holdel, N.J., March 1989.

[32] Richard A. Becker, Stephen G. Eick, Eileen O. Miller, and Allan R. Wilks, "Network Visualization," *24th Int'l Symp. on Spatial Data Handling Proceedings*, Zurich, Switzerland, pp. 285-294, July 1990.

[33] Ben Shneiderman, "Dynamic Queries for Visual Information Seeking," *IEEE Software*, vol. 11, no. 6, pp. 70-77, Nov. 1994.

[34] Christopher Ahlberg and Ben Shneiderman, "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays," *CHI 94 Conf. Proceedings*, Boston, Mass., pp. 313-317, April 1994,

[35] Jerome H. Friedman and Lawrence C. Rafsky, "Fast algorithms for multivariate lining and planning," *Proc. of Computer Science and Statistics: 8th Annual Symp. on the Interface*, pp. 124-136, 1979.

[36] William S. Cleveland and Robert McGill, "The Visual Decoding of Quantitative Information on Graphical Displays of Data," *J. of the Royal Statistical Soc., Series A*, vol. 150, pp. 195-229, 1987.

[37] Robert Sekuler and Randolph Blake, *Perception*, Third ed., New York, N.Y.: McGraw-Hill, Inc, 1994.

Richard A. Becker is a member of the technical staff in the Statistics Research Department at AT&T Bell Laboratories, Murray Hill, N.J. His research interests are statistical computing, data visualization, and data analysis. He is author of *The New S Language* and a Fellow of the American Statistical Association.

Stephen G. Eick is the Technical Manager of the Data Visualization Research group at AT&T Bell Laboratories. His research focuses on extracting the information latent in large databases using novel interactive visualizations. This involves inventing the techniques, developing the software tools, and building an infrastructure to mine knowledge from corporate databases so that it can be put to competitive and commercial advantage. His research group has developed a suite of visualizations including tools for visualizing geographic and abstract networks, software source code, text corpora, log files, program slices, and relational databases, among others.

Eick is an active researcher, is widely published, and holds several software patents. He is particularly interested in visualizing databases associated with large software projects, networks, and building high-interaction user interfaces. His educational background includes a BA from Kalamazoo College (1980), MA from the University of Wisconsin at Madison (1981), and his PhD in Statistics from the University of Minnesota (1985).

Allan R. Wilks is a researcher in the Statistical Models and Methods Department of AT&T Bell Laboratories. His research interests are in statistical computing, including user interfaces, algorithms, computational geometry, and interactive graphics. He received his BSc in mathematics from the University of Toronto in 1976 and his PhD in statistics from Princeton University in 1980. He is a member of the American Statistical Association.