Supporting the Visual Analysis of Dynamic Networks by Clustering associated Temporal Attributes

Steffen Hadlak, Heidrun Schumann, Clemens H. Cap, Till Wollenberg

Abstract—The visual analysis of dynamic networks is a challenging task. In this paper, we introduce a new approach supporting the discovery of substructures sharing a similar trend over time by combining computation, visualization and interaction. With existing techniques, their discovery would be a tedious endeavor because of the number of nodes, edges as well as time points to be compared. First, on the basis of the supergraph, we therefore group nodes and edges according to their associated attributes that are changing over time. Second, the supergraph is visualized to provide an overview of the groups of nodes and edges with similar behavior over time in terms of their associated attributes. Third, we provide specific interactions to explore and refine the temporal clustering, allowing the user to further steer the analysis of the dynamic network. We demonstrate our approach by the visual analysis of a large wireless mesh network.

Index Terms—Dynamic networks, visualization, supergraph clustering

1 Introduction

The visual analysis of dynamic networks is an important subject in many fields concerning for example social or citation networks, power grids or communication networks. Dynamic networks can be described as a sequence of networks, each consisting of a set of nodes and edges. With both, nodes and edges, attributes can be associated that also vary over time. An important task often encountered in their analysis is the identification of temporal relationships between nodes and edges. More precisely, the analysts are interested in finding groups of nodes and edges featuring similar trends of their attributes over time. Finding these groups allows them to determine stable, evolving or unreliable parts of these networks which is for instance of importance for the maintenance in communication networks or discovering evolving topics in citation networks.

Common approaches for visualizing dynamic networks utilize animations [9, 13], small multiples [4] or layering to convey the dynamics of these networks [5, 6, 7, 25]. Larger dynamic networks are often abstracted by clustering the nodes and edges for each time step separately [12, 11, 18]. While providing appropriate insights into the structural changes, these approaches do not address the associated time series sufficiently. The attributes of every time point have still to be analyzed by the user step by step, which leaves him with the burden of manually identifying temporal relations communicated with different images.

Furthermore, approaches reducing the temporal domain are often based on calculating a supergraph. A supergraph represents the union of all networks in the sequence of the dynamic network and is thus also referred to as union graph. In this way, it summarizes the dynamic network containing all nodes and edges that were present at least for one time step. On this basis, these approaches can provide an overview of the structure of the dynamic network in only one image but for the price of almost completely removing the temporal information [27, 23]. In the case of smaller networks, additional temporal information is included as small time plots embedded into the network visualization showing a single attribute over time [26, 28, 33]. However, in this way, discovering nodes and edges featuring a similar course over time is a difficult endeavor.

That means, the first group of approaches adequately communicates structural changes but requires a manual matching of time series that are presented by different images. In contrast, the second group of approaches provides only one image but shows a summary of nodes, edges and attributes that does not support the analysis of a particular period of time. With this paper, we make a first step in closing this gap.

Our goal is to support the analyst in finding and analyzing substructures sharing similar attribute trends over periods of time. Based on the supergraph of the dynamic network, we therefore introduce a temporal clustering with regard to the associated time-varying attributes. This clustering is based on adapting and applying a clustering for time series data which has been successfully used for instance for the analysis of spatio-temporal data [2, 19, 34]. The resulting clustering of the associated time series is then used to group the corresponding nodes and edges of the supergraph. The visualization of this clustered supergraph communicates both a *structural overview* of the discovered groups of nodes and edges with similar trends of associated attributes as well as a *temporal overview* of these trends. Additionally, we provide methods to refine the temporal clustering allowing the analysts to adapt it to their needs such as restricting the time interval or selecting the attributes to be analyzed.

The paper is organized as follows: The temporal clustering is described in Section 2, followed by a description of our visual design and appropriate interactions in Section 3. In Section 4, our approach is demonstrated by applying it to a data set obtained from a wireless mesh network consisting of 290 nodes and 1,888 edges captured over 21,733 time points (every 10 minutes) spanning 5 months of data. All figures shown throughout the paper are based on this data set. We finally conclude our paper in Section 5 by discussing future work.

2 Temporal Clustering of Dynamic Networks

Our aim is to support the analyst in identifying and inspecting groups of nodes and edges that are similar according to their attribute values over time. For the automatic extraction of these substructures, we cluster the supergraph by utilizing techniques stemming from the analysis of time series data. However, before these techniques can be applied, we have to solve the following problems:

1. The attributes for all nodes and edges have to be extracted to define the time series.

2. Their clustering has to be configured and carried out.

3. Based on the clustering, connected subgraphs have to be defined.

These three aspects are described in the following.

- *Steffen Hadlak, Heidrun Schumann, Clemens H. Cap and Till Wollenberg are with the University of Rostock, E-mail: firstname.lastname@uni-rostock.de.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 27 September 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

2.1 Extraction of Time Series

The time series to be clustered are associated as attributes with nodes and edges of a given dynamic network. As not all attributes may be of interest, first those attributes have to be selected that should be used for the clustering. The supergraph contains all nodes and edges of the dynamic network. Hence, we can extract for each of these nodes and edges a time series representing the values of each selected associated attribute.

As nodes and edges may not exist at all time steps, we have to mark missing values by a particular value. A common way is the missing-indicator method introducing a placeholder value for each attribute that is not within the range of existing values. Thus, this placeholder is used to differentiate between missing and original values [10]. Yet, it has to be used with care for the clustering as it may lead to a biased analysis. For instance, two nodes that coexist for a short amount of time and are absent at the remaining time points will be grouped together because they show the same placeholder value most of the time. This grouping is rather independent of the real values they feature during the time they exist. An alternative to compensate for such a problem is to neglect time points at which at least one node does not exist. In this way, those nodes would be grouped with other nodes or edges sharing a similar course at only those time points they coexist.

However, both approaches yield interesting information. The first allows for the grouping of coexisting subgraphs and thus the determination of parts of the network that have been deleted or created at the same time. Whereas the second approach allows for the analysis of nodes and edges according to the behavior they show during the time they exist. In communication networks, this could be used to determine stable connections independently of their creation date. So this leads to another parameter that may be used to steer the analysis.

Furthermore, to filter out minor fluctuations such as unwanted noise and to allow for an interactive analysis by decreasing the calculation time of the clustering, often a reduction of the time series is performed. The reduction we apply is based on a simple low-pass filtering averaging the data values of multiple time steps and a subsampling resulting in a coarser and smoother temporal scale to be clustered as proposed in [29]. A more complex alternative is to extract perceptually important points to represent the main characteristics of the time series [34]. Yet, its non-uniform distribution of time points complicates the similarity calculation during the clustering.

2.2 Clustering of the Supergraph with regard to the associated Time Series

Clustering time series requires two steps: determining a distance/similarity measure and selecting a method for clustering the time series based on this measure [30]. Measure and method can be chosen and configured independently of each other, enabling different ways to analyze the dynamic network. Currently, we provide three different similarity measures: the Euclidean distance, the correlation coefficient and a trend-based similarity measure [19] as well as two clustering methods: a hierarchical and a k-means method.

The choice of the similarity measure strongly depends on the chosen attributes and on the analysis goals of the user. If he is interested in grouping the nodes and edges with regard to simultaneous changes over time, a trend-based measure would suffice. For example, this can be the case if he is looking for evolving topics in a co-authorship or citation network which is characterized by a simultaneous increase in the node degree or citation count. Yet, if he is interested in nodes and edges showing similar values over time, the Euclidean measure is better suited. This is generally a good first choice for attributes featuring only a single data range and that are thus comparable like the quality of connections between nodes in a communication network. To be able to compare different similarities these measures are normalized to the range [0,1] with 0 describing no similarity of the time series and 1 describing full similarity.

As a default clustering method, we use a hierarchical clustering method as it supports an interactive exploration of the network on multiple levels of detail. In case of a k-means clustering, the number and center of clusters have to be appropriately specified.

2.3 Defining connected Subgraphs

The result of the clustering is solely based on the time series extracted for the attributes. Thus, structural connections between nodes and edges are not considered. Furthermore, attributes may be either defined for nodes or edges. That means that each cluster represents a collection of possibly disconnected nodes or edges. However, each cluster can be transformed into a subgraph as follows:

1. Extraction of the induced subgraph: In case of a cluster containing disconnected nodes often as a result of attributes defined only for nodes, we add all edges whose incident nodes are both contained in the cluster. For a cluster containing disconnected edges which may arise from attributes defined only for edges, we add the incident nodes.

2. Path-preserving clustering: The induced subgraph may still consist of multiple components that are similar according to their time series, but that are not structurally related. This can be exemplified by a simultaneous drop-out of nodes within a communication network. If these nodes are clustered together but are not connected, their drop-out may be a coincidence. However, if these nodes are connected, it is more likely that their simultaneous drop-out is caused by the same incident. Thus, to obtain a clustering in which each cluster represents a single connected subgraph (also called "path-preserving" clustering [3]), all discovered clusters are split into connected components afterwards. If the analyst is interested in finding relations that are not covered in the structure but appear in the temporal clustering, the maintenance of the path-preserving property can also be skipped making this step optional depending on the analysis.

The clustered supergraph resulting from these steps is used in the following section for the visualization of the dynamic network.

3 Visualization of the Clustered Supergraph

The goal of our visual design is to support the visual exploration of a dynamic network. We therefore provide an overview and several detail views. The supergraph provides the overview showing groups of nodes and edges being similar with regard to the temporal trends of their associated attributes. Clustering results and time series of interest are represented by detail views. Figure 1 demonstrates our design. In the following, we will describe the views in more detail.

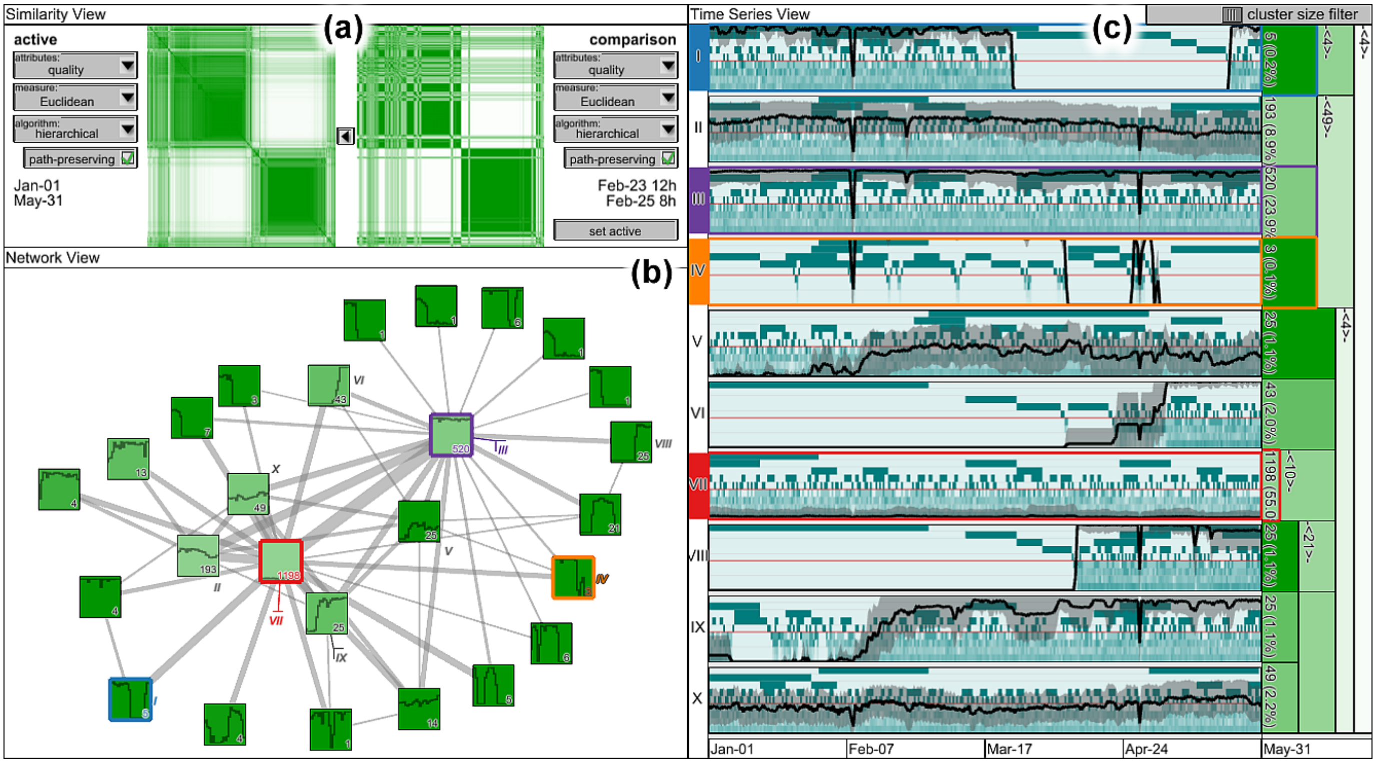
3.1 Network View

Visual Design: The *network view* (see Figure 1b) visualizes the clustered supergraph. The temporal trend of each cluster is represented by small glyphs as suggested by [24, 28]. These glyphs show small time plots for a selected attribute. The similarity of the time series of each cluster (intra-cluster similarity) is mapped to the background color of its glyph. High similarity is represented by dark green and low similarity by white glyphs. The size of each cluster is reflected by a label in the lower right corner of its glyph. Furthermore, to reflect the structural aspect of the dynamic network, clusters containing nodes or edges that are connected within the network are connected by lines. Here, the width of these lines reflect the number of connections between the clusters.

To sum it up, the *network view* communicates the connectivity and intra-cluster similarity of clusters as well as the major temporal trend of each cluster.

Interactions for Exploring the Structure: In the case of a hierarchical clustering, we utilize a layer-based approach. The *network view* then represents a single layer in the clustering hierarchy. We provide a layer-based navigation allowing the user to select a similarity threshold and thus the layer to be visualized as well as an unbalanced drill-down by expanding and folding specific clusters on demand.

The user can also select multiple clusters for a closer inspection. These clusters are highlighted in the *network view* as well as in the following *time series view* by a colored border. Therefore, we provide a couple of distinguishable colors (e.g., qualitative color schemes in [15]) to differentiate the selected clusters and allow their localization in both views. For each cluster, the contained substructure is



active

attributes: quality

measure: Euclidean

algorithm: hierarchical

path-preserving ☑

Jan-01
May-31

comparison

attributes: quality

measure: Euclidean

algorithm: hierarchical

path-preserving ☑

Feb-23 12h
Feb-25 8h

Fig. 1. Our visual design for the analysis of a clustered supergraph. The similarity view (a) shows two clustering settings: an active one which serves as a basis for visualizing the graph and an additional one with possibly changed parameters. The associated similarity matrices demonstrate to which degree the clustering would change for the second parameter setting. The network view (b) provides an overview of the clustered supergraph. Nodes represent clusters abstracting subgraphs with a similar behavior over time and edges reflect connections between these clusters. Nodes are visualized as glyphs showing small time plots that represent the temporal trend of the associated cluster as well as its color-coded intra-cluster similarity. The temporal trends are visualized in more detail in the time series view (c) by a multi-scale time plot for each visible cluster in the network view. The cluster hierarchy is shown on the right side of this view, also reflecting the intra-cluster similarity. Clusters (*I*, *III*, *IV* and *VII*) selected for a closer inspection are highlighted in different colors in the network and time series view.

drawn in the according color in the background of the *network view* as shown in Figure 8 allowing for a quick peek into the details of multiple clusters on demand.

3.2 Time Series View

Visual Design: Each cluster visible in the *network view* is represented by a time plot in the *time series view* (see Figure 1c). The time plot shows either the average trend of the time series or the cluster representative. As the time series is often larger than the available amount of pixels, the data is typically shown on a coarser scale and thus details may be hidden. To support the user in finding those details even at finer scales, we utilize the idea of a multi-scale time plot as introduced in [21]:

First, for each time series a number of coarser scales are extracted for instance by iteratively applying the reduction described in Section 2.1. Between each pair of subsequent temporal scales, the heterogeneity is determined for example by calculating the differences in the slopes. This heterogeneity is then visualized as a color coded-band with dark regions representing high heterogeneity. This is shown on the left side of Figure 2. The background of the image is then constructed by stacking these bands with the coarsest scale on top. In this way, the background provides pointers to time intervals with more information than visible on coarser scales. For the visualization of the time series, a temporal scale with a resolution fitting the number of available pixels is chosen. This scale is communicated by a red line between the bands including this scale. The resulting multi-scale time plot is depicted on the right side of Figure 2.

Additionally, the standard deviation is visualized as a semi-transparent ribbon along the time series to communicate the similarity within the cluster over time. This allows the user to discern between clusters that are similar across the whole time series and clusters featuring a high dissimilarity only for a few time points.

With a hierarchical clustering, the time plots are accompanied by an *icicle plot* [17] showing the clustering hierarchy up to the selected layer. Each cluster is represented by a colored rectangle representing its intra-cluster similarity. Here, the same color-coding is used as in the *network view* by mapping high similarity on a dark green color and low similarity on a white color. For those clusters currently visible in the *network view*, the corresponding rectangle also provides the number of nodes and edges it contains as an absolute number and as the percentage of the whole graph in brackets.

In this way, the *time series view* provides insight into the temporal trends of each cluster as well as visual hints pointing to temporal scales on higher granularity levels or time points that may provide additional information. Moreover, together with the *icicle plot*, they show the similarity within each cluster.

Interactions for Exploring the Time Series: Besides zoom and pan for inspecting details in the time plots, we provide filtering mechanisms to cope with a high number of clusters shown in the *network view* allowing to visualize only the time series for those clusters that are most important to the user. For this purpose, he can either select a minimum number of nodes and edges a cluster has to contain or directly select the clusters to be analyzed in the *network view*. The icicle plot provides visual feedback on the filtering by showing the number of nodes and edges that were filtered in angle brackets.

3.3 Similarity View and Visual Comparison of Clustering Settings

The presented views communicate the information captured by the clustered supergraph. Yet, different clusterings that consider for in-

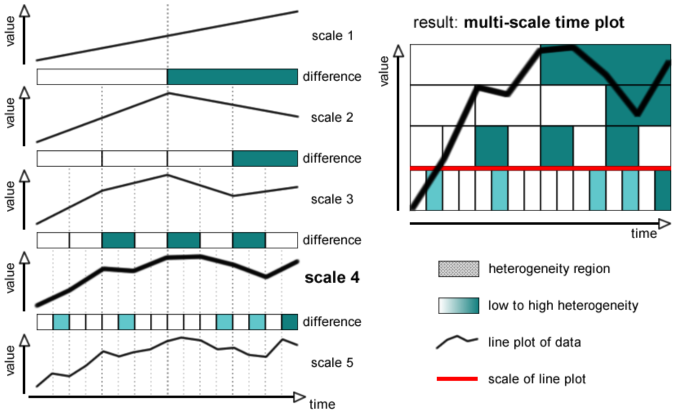
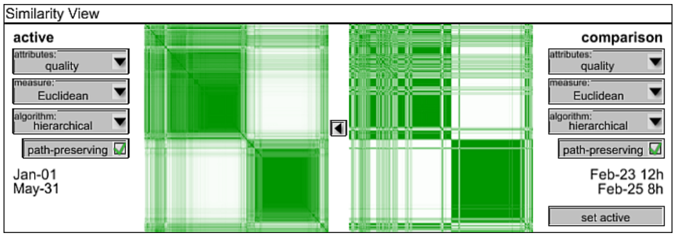


Fig. 2. The construction of a multi-scale time plot according to [21]. Left side: Different scales of the time series. Between subsequent temporal scales, the heterogeneity is calculated and visualized as a color-coded band. Dark regions represent high heterogeneity and thus communicate high differences between the scales. Right side: These bands are stacked to form the background of the plot providing pointers to time intervals with more information. The scale above the red line is used to represent the time series.


Similarity View

active
attributes: quality
measure: Euclidean
algorithm: hierarchical
path-preserving ☑
Jan-01
May-31

comparison
attributes: quality
measure: Euclidean
algorithm: hierarchical
path-preserving ☑
Feb-23 12h
Feb-25 8h
set active

Fig. 3. The similarity view shows the active clustering settings and the corresponding similarity matrix on the left side and the settings and matrix for a second clustering on the right side. High similarity of time series is mapped to dark green, low similarity to white. The second clustering was restricted to a smaller time interval (see Figure 7c for its selection). This resulted in a split of a large cluster visible from the breaks on the right side in the formerly homogeneous block.

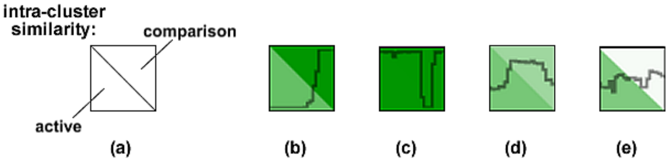


Fig. 4. Adaptation of the glyphs for representing clusters to accommodate two clustering settings: (a) The glyph background for representing clusters is split into two parts visualizing the intra-cluster similarity according to the selected similarity measure of both clustering settings. High similarity is mapped to dark green, low similarity to white. (b) and (c) exemplify a cluster that will remain after the switch, while (d) and (e) show a possible split of the cluster.

stance different attributes or are based on different measures would typically offer further information. Hence, the user needs appropriate interactions to adapt the clustering to his analysis needs as well as visual cues for estimating the effects of different clustering settings.

Therefore, we allow the user to define and configure two clustering settings at the same time. First, an active one used for calculating the clustered supergraph that is used for the visual exploration in the *network* and *time series views*. And a second one used to provide visual cues of how the active clustering will be affected by changes to clustering parameters. At any time, the user can switch between the two settings allowing him to perceive the dynamic network from different perspectives. For this purpose, we provide a *similarity view*.

Visual Design: The *similarity view* (see Figure 1a) shows two clustering settings and provides controls for their adaptation. It visualizes a similarity matrix for each setting capturing the similarities between all extracted time series according to the selected similarity measure. To communicate possible clusters, the time series are ordered to form homogeneous blocks as described in [32]. This relation is demonstrated in Figure 3 allowing for the identification of two larger blocks as clusters. In this way, the similarity matrix supports:

- with a k-means clustering, the estimation of the number of clusters and the specification of cluster centers.
- with a hierarchical clustering, the extraction of an appropriate layer.

The layout of the matrices is synchronized showing the same order of time series. In this way, similar matrices reflect similar clustering results while different results, e.g., a split of a cluster can be perceived as a break in one matrix compared to the otherwise homogeneous block of this cluster in the other matrix as exemplified in Figure 3.

Adaptation of the Views: While the *similarity view* allows for the comparison of two settings at the same time, the *network view* and *time series view* only support the visual exploration of a single clustering result. Hence, we provide two modes of visualizing both clustering settings:

- Only one setting is visualized at any time. In this case, the *network* and *time series view* are used as they have been described.
- One setting – the active clustering – is used as the basis of the visual exploration but additional visual cues according to the second clustering settings are embedded. For this purpose, the *network view* and the *time series view* visualize the active clustering but are adapted as follows to also provide visual cues of the differences to the second clustering setting.

  In the *network view*, the background of the glyphs used to represent the clusters is split into two parts (see Figure 4a). The lower left half is still showing the similarity within the cluster according to the similarity measure of the active clustering while the upper right half is showing its intra-cluster similarity according to the second clustering setting. The representation of the clusters in the *icicle plot* is handled accordingly. In this way, the similarity matrices as well as the glyphs communicate whether with a new clustering setting a current cluster would still form a cluster (for instance Figure 4b and 4c) or would break up (Figure 4d and 4e). Based on these visual cues, the user can estimate if a new clustering may yield additional information and thus if it is beneficial to calculate and explore a new clustering.

In summary, the *similarity view* provides an overview of two clustering settings and shows their differences. In this way, it allows an informed refinement of different clustering settings. Furthermore, the adapted views communicate changes in the similarity within the clusters. Altogether, all views – the *similarity view* as well as the *network* and *time series view* – provide valuable insight not only for the analysis of a single clustering but also for steering the refinement of the clustering settings.

Interactions for Refining the Clustering: For adjusting the settings, we provide the following functionality:

- choosing between different attributes, similarity measures and clustering methods as well as adapting specific parameters in the *similarity view* (see Figure 6 for adaptations of the number of clusters) allowing for the analysis of correlations between different attributes.
- selecting a subset of nodes and edges in the *network view* to be

clustered according to different settings supporting the analysis of a specific cluster of interest.

- selecting a time interval in the *time series view* (see Figure 7c) for which the time series are extracted to support the analysis of local trends and their comparison to global trends.

Adjustments of the settings of both clusterings are reflected in all views with different effects. Interactions with regard to the active clustering immediately lead to an update of the *network* and the *time series view*. However, adapting the second clustering only leads to an update of the background color of the glyphs in *the network view* and the *time series view*. Therefore, the user can refine this clustering without loosing his current view on the dynamic network, compare the different settings and switch to this clustering if it shows significant changes.

Overall, the three views provide an overview as well as detailed views of a supergraph clustering according to temporal trends of associated attributes. Furthermore, they reflect similarities and differences between two clustering settings. In this way, these views do not only support the analysis of a single clustering result but also support an interactive and informed refinement of the clustering. This allows the analyst to determine global trends affecting the whole network as well as to discover local trends only affecting specific substructures or time intervals and their comparison. How this interplay of clustering, visualization and interactive refinement can support a visual analysis of a real world dynamic network is discussed in the next section.

4 Use Case

Our use case concerns a data set of link qualities in a wireless mesh network. The data was collected in the *OpenNet*[1], a wireless community network located in Rostock, North Germany. The network covers parts of the city and some villages in the surrounding region.

4.1 Data Description

To give an impression of the network structure, Figure 5 shows the OpenNet nodes and links overlaid on a map. Almost all nodes are fixed in place and for the majority of nodes, the geographic position is available as latitude and longitude tuple. It can be seen that the network has some densely populated clusters in the city center and in three villages south of Rostock while it is sparsely populated in other regions. It can also be seen that the clusters are connected with one or more long-distance links.

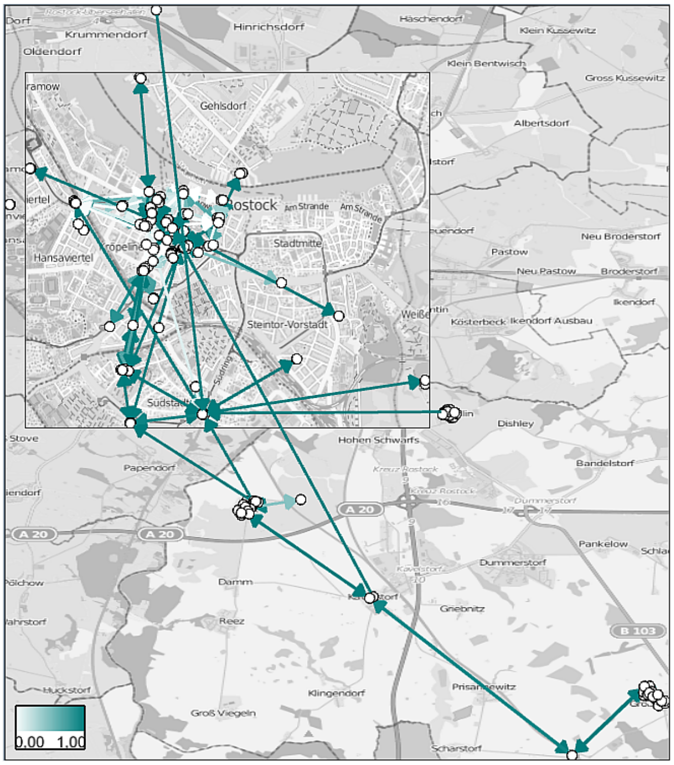


Fig. 5. Overview of the structure of the OpenNet network. As almost all nodes have a fixed geographic position they are laid out on top of a map (© OpenStreetMap contributors). The link quality of their connections (edges) is mapped on the color. Dark represents high quality and light low quality. The region around the city center of Rostock is enlarged providing a more detailed view while maintaining the overall structure following the idea of an In Situ embedding [14]. Densely populated clusters are situated in the city center and three villages south of Rostock while other regions are sparsely populated.

The network is based on IEEE 802.11 Wireless LAN technology and uses an extended version of the Optimized Link State Routing protocol (OLSR, [8]). The original protocol classifies links as either working or non-working. This does not work well for wireless networks where intermediate packet loss rates are perfectly normal. The extended version OLSR-LQ [20] accommodates this by associating a quality value with each link. A link's quality value *LQ* describes the probability of a successful packet transmission on this link. A perfect link, i.e., a link showing no packet loss at all, has an LQ of 1.0. Whereas a poor link, i.e., a link with a high packet loss, has an LQ near 0.0.

OLSR works proactively and belongs to the link-state class of routing protocols. As a result, each node receives information about all existing links and their qualities in the entire network. From the received information, each node builds-up a weighted digraph representing the network with the weights reflecting the link qualities. This graph is then used to compute optimal paths to each node in the network.

For collecting the data set, the current list of known links was collected from a particular node within the OpenNet every 10 minutes over a five-month period. The data collection took place from January 1st 2011 until May 31st 2011. During this time, the network graph contained on average 174 nodes and 830 edges. The supergraph of all 21,733 recorded snapshots contains 290 nodes and 1,888 edges. For 139 time points, no data is available due to outages of the recording node. The data set is described in more detail in [31] and is publicly available[2].

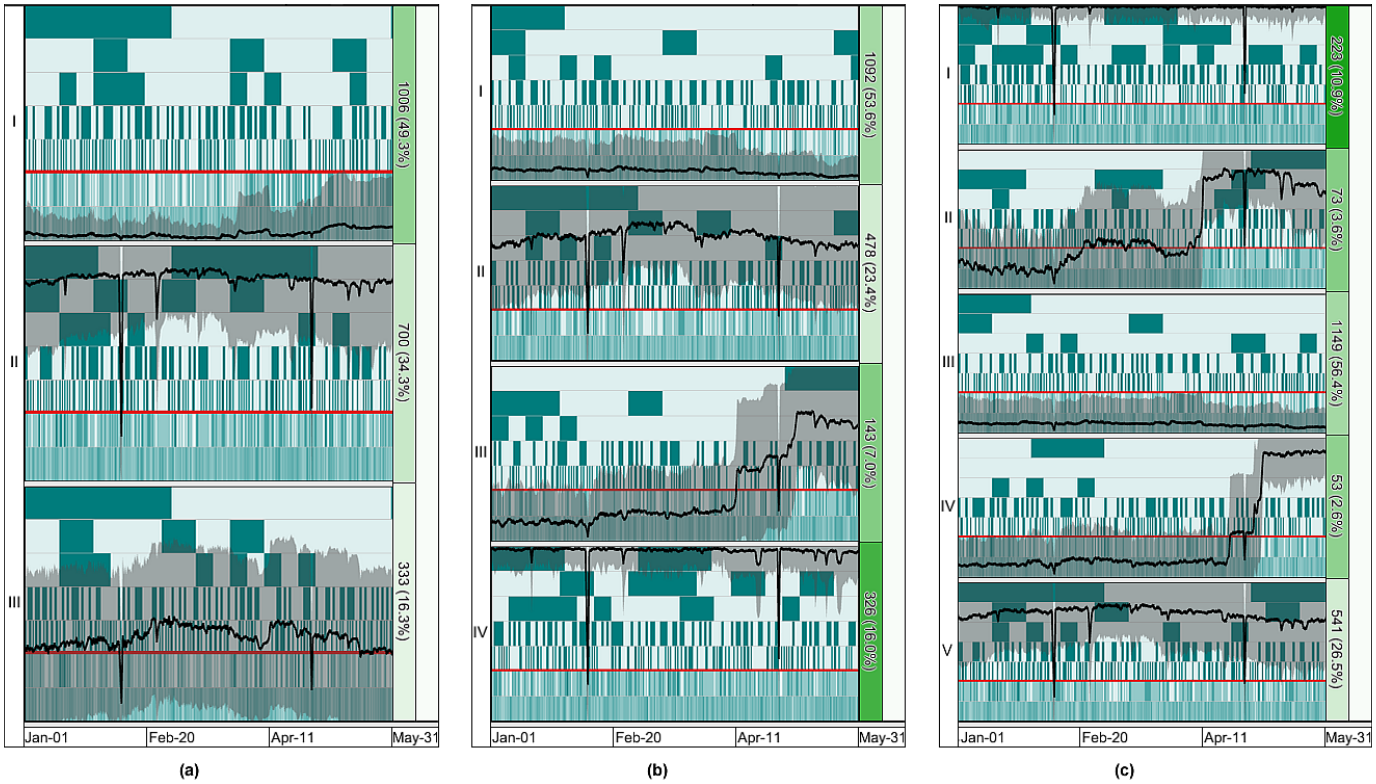
In the original data set, existing links (edges) are described explicitly along with their respective link quality. If a link was not working at a particular point in time, it is missing in the respective snapshot. However, if this link was working at some time during the data collection period, it is included in the supergraph. In our case, it is convenient to have an explicit statement about the quality of any particular link at any particular point in time. Therefore, if a link from the supergraph is missing in a particular snapshot, we add the respective link and associate a zero quality with it, which is semantically equivalent to a non-existing link.

In general, each link has a basic quality magnitude mainly determined by factors such as distance and antenna type. Beyond that, the link quality varies over time. These fluctuations may be caused by interference originating from other radio transmitters, strong load on the respective link or weather effects (e.g., increased attenuation due to moisture or antenna misalignment due to wind).

When a particular link within a path deteriorates, the OLSR software normally selects an alternative path automatically. However, if no alternate links of good quality are available, a sudden strong deterioration of a single link may limit the overall connectivity of entire network segments.

[1] More information in German language available at http://www.opennet-initiative.de

[2] http://opsci.informatik.uni-rostock.de/index.php/Network_quality_measurements



(a) (b) (c)

Fig. 6. The temporal trends for three different clustering settings based on k-means. (a) k = 3: The averaged link qualities support the assumption of three distinct quality levels with cluster *I* containing poor links (low values), cluster *II* good links (high values), and cluster *III* average links (average values). Clusters *II* and *III* exhibit a low intra-cluster similarity which implies that they contain divergent time series, which is confirmed by switching to a larger k. (b) k = 4: The intra-cluster similarity of all clusters increases and a new trend arises. Cluster *III* seems to contain nodes and edges that belong to a part of the network that was connected to the core network in April 2011 for the first time. (c) k = 5: A further increase of k improves the overall intra-cluster similarity. Yet, still some clusters with a high intra-cluster dissimilarity remain. The previous mentioned cluster *III* split into the new clusters *II* and *IV*.

4.2 Analysis Goals

The general goal of the OpenNet members is to maintain a network quality satisfying all users. Therefore, they need to identify low quality links that may cause low network performance. Considering the number of links and time steps of the recorded data set, this may become a challenging task. In general, the OpenNet members are interested in analyzing the data on multiple levels:

- They want to monitor the network status on a *global level* and identify general problems early.
- They also want to localize the source of sporadic problems and determine which parts of the network were affected by them. In this case, their focus is on a *local level*.

Both goals are addressed separately in the following two sections.

4.3 Analysis on a Global Level

In order to solve the first problem, the OpenNet members want to identify which links persistently show a high quality, which links are fluctuating, and which links show a constantly low quality. Commonly, they would have to analyze each time point separately to filter out those links that frequently have a low quality or links that exhibit sporadic shifts in quality.

With our approach, the goal can be reached much easier. At least one attribute and a similarity measure have to be chosen to extract the nodes and edges the user is looking for, i.e., the nodes and edges that exhibit a similar behavior. As the user is interested in grouping them according to their connection quality over time, the Euclidean distance measure calculated for this attribute is a good choice.

The network monitoring and graphic tools currently used within the OpenNet distinguish three levels of reliability of the links (good, average, poor) based on their link quality. We therefore start our analysis with a k-means clustering method and choose an initial $k = 3$. This choice is also supported by looking at the *similarity view* depicted in Figure 1a. There we can see two large homogeneous blocks representing two distinct clusters and a few smaller ones.

The time series of the three resulting clusters are shown in Figure 6a. The averaged link qualities support the initial assumption of three distinct quality levels. Cluster *I* exhibits low LQ values corresponding to poor links, cluster *II* shows high LQ values and thus captures good links, and cluster *III* contains average links. However, clusters *II* and *III* exhibit a low intra-cluster similarity as can be seen from the rectangles on the right side showing light colors. Their intra-cluster dissimilarity implies that they contain more, divergent time series.

This implication is confirmed by choosing different, larger k as depicted in Figures 6b and 6c for $k = 4$ and $k = 5$. With increasing number of clusters, the intra-cluster similarity of all clusters rises. While the “good” and “poor” clusters remain, the divergent time series are captured in the emerging clusters.

The clusters *III* in Figure 6b as well as *II* and *IV* in Figure 6c seem to contain nodes and edges that belong to a part of the network that was connected to the core network in April 2011 for the first time. This can be seen from the fact that the elements of both clusters show a constantly low (close to zero) LQ value until April while their average LQ value is high and shows only few fluctuations afterwards.

Even with increasing k, some clusters remain heterogeneous. So, by switching to the hierarchical clustering method, instead of specifying the number of clusters to be extracted, we can define the level of intra-

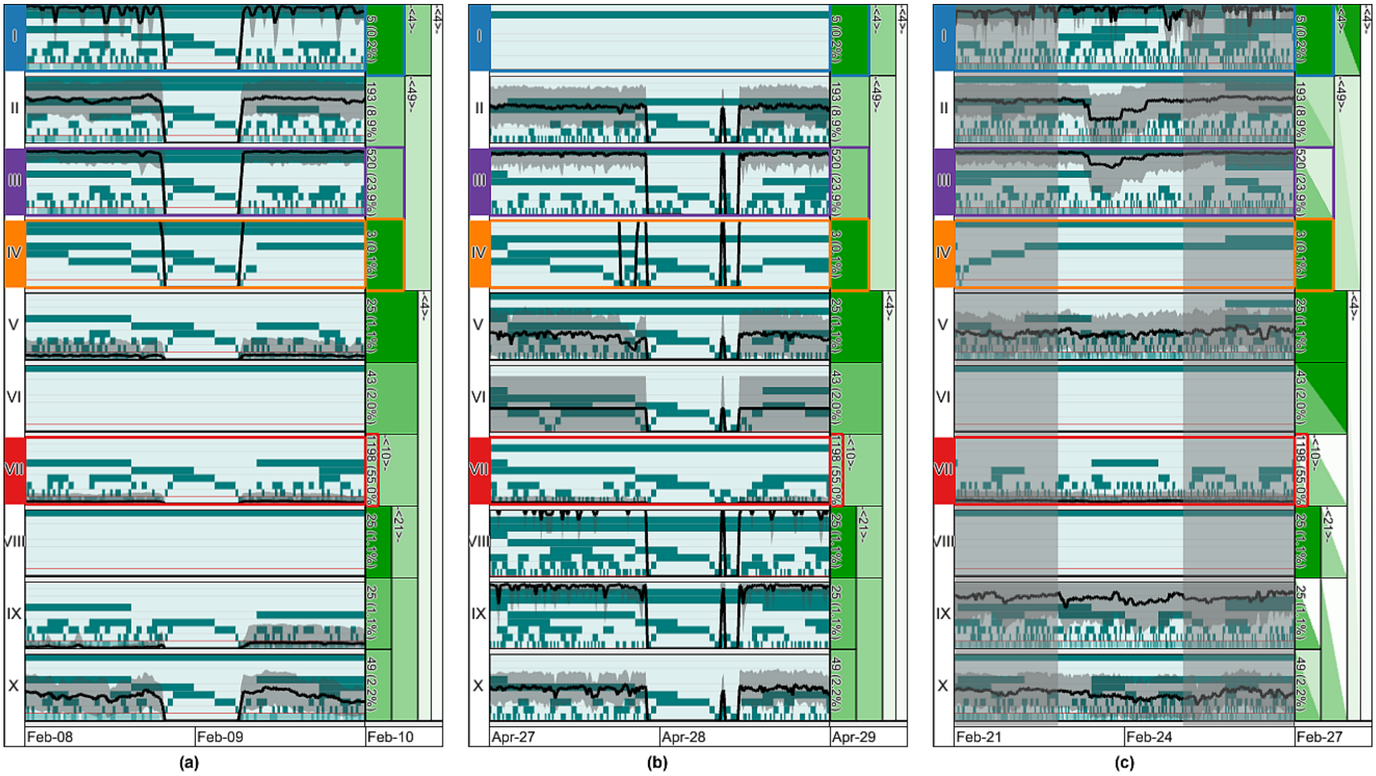


Fig. 7. A zoomed-in view of three time intervals showing drop-offs in the quality of clusters with a generally good quality otherwise. (a) and (b) show a global drop-off of all clusters simultaneously. In both cases the recording node was disconnected and thus these intervals contain no data at all. (c) on the other hand shows a larger drop-off only in the clusters *II* and *III*. Its cause has to be a local event. A time interval around this event is selected for closer inspection (time points outside of this interval are faded-out in gray). The icicle plot on the right reveals that the intra-cluster similarity of both clusters in the selected time interval is lower (light color of the upper right triangle) than for the complete interval (green color of the lower left triangle). If a new clustering is calculated on the basis of this time interval the degradation of their intra-cluster similarity will lead to a split of these clusters into multiple components.

cluster similarity we want to achieve. The result of the hierarchical clustering is shown in Figure 1. Generally, we can differentiate two groups of clusters:

- The first group shows a low variance in their LQ values over time. Yet, each cluster has a distinct quality level. This group contains the clusters *II*, *III*, *VII* and *X*. Here, cluster *III* captures the steadily good links, clusters *II* and *X* the average links and cluster *VII* the constantly poor links.
- The second group contains clusters showing a higher variance over time, i.e., they exhibit different quality levels at different time intervals. This group consists of the clusters *I*, *IV*, *V*, *VI* and *IX*. For example, clusters *VI* and *IX* capture the part of the network that was connected to the core network in April 2011. As the links were not present until April 2011 they show a constant quality of 0.0. Yet, after they were established, they exhibit generally high LQ values and thus fall into the good quality level.

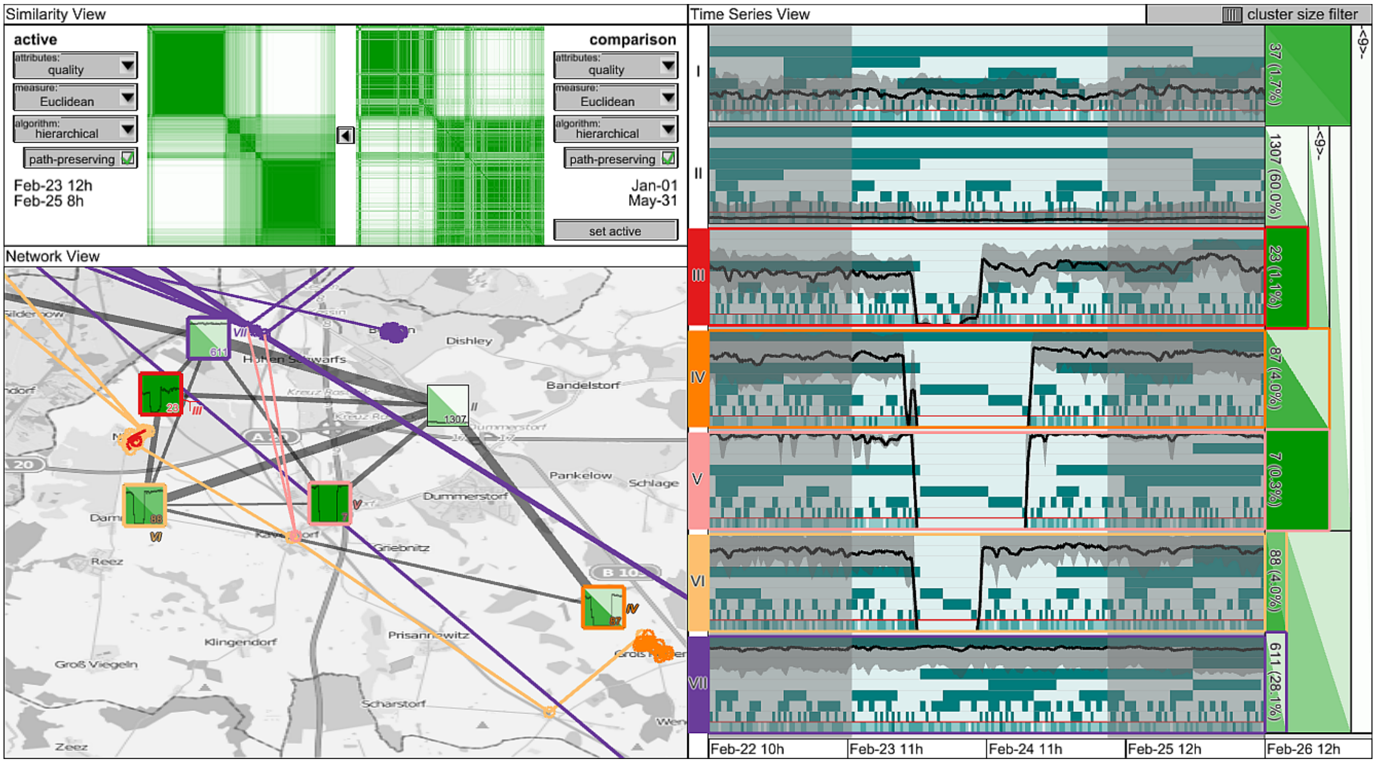
Both groups describe interesting features to be analyzed further. However, the second group represents only a minority of nodes and edges (as indicated by the labels in their glyphs and in the icicle plot) and is therefore of lesser interest for the OpenNet members. In contrast, the majority of nodes and edges are located in the first group. When looking at the clusters *II* and *III*, we see that one third of all nodes and edges exhibit a constantly good or high quality while the cluster *VII* shows that nearly 50% of all nodes and edges in the network have a consistently low quality.

The large amount of low quality edges does not imply a low network quality in general because a weak link only has an effect on the network quality if there are no better links available to reach a certain destination. Weak links have a significant impact if they are the only connections between substructures in the network. The impact is particularly strong if the connected substructures have a similar high quality otherwise. In order to distinguish weak links with significant impact on the network from links which are weak but redundant, the path-preserving clustering is very useful. During the path-preserving clustering, this constellation results in a split of the cluster containing these substructures into multiple components.

The four selected clusters in Figure 1 illustrate the different cases. In the *network view*, the blue colored cluster (*I*, lower left corner) is connected to another cluster with high quality. Yet, both are connected to two other clusters that have a constantly low quality. Further investigation shows that these two clusters contain three nodes which are located in the west of the city center at the border of the OpenNet network. With this information at hand, OpenNet members could try to improve the links that connect these nodes with the remaining network. We can also see an orange colored cluster (*IV*, lower right corner) that is connected to the low-quality, red cluster (*VII*). However, this cluster is likewise connected to a cluster of nodes and edges that show a constantly high quality (*III*, purple color). Thus, the periods of low quality in this orange colored cluster cannot be explained solely by the quality of the intermediate clusters but require further investigation.

4.4 Analysis of Local Problems

Based on the global overview described in the previous section, local events in the different clusters can be identified. This identification is important to address the second problem which concerns sporadic drops in the quality of important links or nodes. Links and nodes are important if their outage would lead to a separation of parts of the network from the remaining core network. Due to the structure of



active
attributes: quality
measure: Euclidean
algorithm: hierarchical
path-preserving ☑
Feb-23 12h
Feb-25 8h

comparison
attributes: quality
measure: Euclidean
algorithm: hierarchical
path-preserving ☑
Jan-01
May-31

Fig. 8. The supergraph is clustered according to the time interval as selected in Figure 7c. The time points outside of this interval are faded out in gray. The clusters are laid out on top of a map (© OpenStreetMap contributors). Cluster *VII* which shows a generally good link quality is highlighted in purple and the clusters (*III*, *IV*, *V* and *VI*) showing the drop-off are highlighted in red to orange. The network view reveals that these four clusters completely contain three connected villages dropping out at the same time. This was caused by a power black-out in that region.

the OpenNet, such effects usually affect local subgraphs. In this context, the second group of clusters seems interesting because of the high variance in their LQ values over time. However, as this group contains only a small number of nodes and edges, their influence on the overall functionality of the network is limited. Hence, this group is of lesser importance for the OpenNet members. They are more concerned about the clusters *II* and *III* of the first group showing a steadily average to good LQ over time. In this way, these two clusters describe the core network and a drop in their quality can have a more severe effect on the network.

By looking at the clustering as shown in Figure 1c, we can see that despite their generally good quality both clusters show drops in the quality that are interesting for a further inspection. In the *time series view*, we zoom in on the three largest peaks visible in clusters *II* and *III* (see Figure 1). The result of the zoom operation is shown in Figure 7. The peaks were caused by two different incidents. The two largest peaks are the result of a disconnection of the data recording node and thus only point to missing data. This is visible in Figure 7a and 7b as this event has a global effect on the time series of all clusters. Yet, there is a peak visible for only two clusters in February that is not caused by the data gap and thus shows a more local incident.

Next, we calculate a new clustering of the network by selecting the time interval around the peak (Figure 7c) in order to narrow down the cause of the peak. The result of this refinement is shown in Figure 8. Looking at the *similarity view* and the *time series view*, we can see that the peak was caused by a subset of nodes and edges that completely lost their connection while other parts of the OpenNet remained functional. Clusters *III*, *IV*, *V* and *VI* (colored in orange to red) consist entirely of disconnected network elements. The *network view* (Figure 8) reveals that these four clusters contain nodes and edges located in three villages south of Rostock. Further investigation revealed that all three villages were disconnected from the core network on February 9th due to a power outage which affected some nodes that connected the villages with the city network (cluster *VII* highlighted in purple).

4.5 Use Case Summary & Discussion

In this section, we have demonstrated how our novel visualization approach supports the analysis of the OpenNet data set. The first discussed goal concerns a global overview of the dynamics within the network. Traditionally, in order to reduce the burden of analyzing all time steps separately, a common approach would be to look at the supergraph only. Therefore, the time series data from each node and edge would need to be mapped to a few attributes that can be visualized in parallel. For example, the mean and the standard deviation of a link's quality could be mapped to color and style attributes of a line. In practice, this approach is problematic because time series mostly cannot be described adequately by a small number of attributes. In Figure 1b, we can see an example of this problem. Here, clusters *I* and *IX* show an almost identical mean and a similar standard deviation while it can be easily seen that their behavior over time is quite different. In contrast, our approach shows a clustered supergraph as a compact overview that helps to identify problematic links without burdening the user with analyzing all time steps separately and without losing important features of the corresponding time series.

The second analysis goal involves the identification of sporadic changes in the quality of the clusters such as temporary outages. A traditional approach to identify such outages would be to look for significant drops in the quality in the individual time series of every node and edge. For assessing the scope of an outage, the user would then need to compare the different time series to find nodes and links with similar behavior. Considering the large number of links within the OpenNet this would be a time-consuming task. Alternatively, animations could be used to identify sudden drops in link quality. However, when analyzing large time spans this would become time-consuming as well. Additionally, once a sudden change is found, the user would have to find out manually if the problem reoccurs at other time steps. With our approach, identifying sporadic, strong changes of link qualities and estimating their scope becomes much easier. The interactive refinement of clustering settings and time ranges allows for an efficient analysis of sporadic drops in the link quality that captures both structural and temporal locality of the phenomena.

The presented visualization approach has supported the OpenNet members in the analysis of their dataset by simplifying the problem of identifying nodes and edges sharing a similar temporal behavior while considering the overall network structure. While these results are promising, we are also looking into other dynamic networks to further evaluate our approach as discussed in the following section.

5 Conclusion and Future Work

In this paper, we have presented the idea of a temporal clustering for large dynamic networks that groups nodes and edges by the temporal trend of their associated attributes. An appropriate visual design based on the clustered supergraph provides an overview of the dynamic network as well as detailed information of the groups and their temporal trends. The integration of visual cues communicating the effects according to two clustering settings does not only allow for showing the clustered dynamic network, but also for estimating the information yield when adapting clustering settings. Coupled with specific interactions, the visualization allows for an exploration as well as an interactive and informed refinement of the clustering. Altogether, this interplay of computation, visualization and interaction supports the user in discovering and analyzing global and local trends affecting different parts of the network. A first use case study for a wireless communication network with data captured over a long period of time has shown its usefulness and its advantages over existing techniques with regard to analyzing the associated attributes over time.

As common approaches for providing an overview of the dynamic network are often based on the visualization of the supergraph, they normally cannot appropriately describe the temporal trends of associated attributes because of the number of nodes and edges to be shown. With our approach, we are able to reduce the amount of nodes and edges to be visualized allowing us to provide time plots showing the temporal trends of their attributes over time. Additionally, the clustering and the visualization of the major trends for each cluster enables the user not only to discover events that are influencing the whole network but also local substructures of the network. This is often a difficult and time-consuming endeavor with existing techniques such as animations or small multiples because of the number of nodes and edges as well as their associated time series that have to be compared manually.

While we have demonstrated our new approach by the visual analysis of a specific use case, we are currently inspecting its applicability for a more diverse set of use cases. In a co-authorship or citation network, evolving topics can for instance be discovered by clustering the nodes and edges with regard to their coexistence over time. This is feasible by extracting an appropriate time series for the coexistence of nodes and edges and thus allows not only for the clustering according to attributes over time. The clustering separates groups of nodes and edges that were already connected over a longer period of time and groups sharing connections only very recently. The latter case corresponds to newer or evolving topics. Another example is the analysis of discussion threads in which edges describe discussions between two or more persons. Here, the clustering can help to discern between short term and long term discussions.

In this context, we also like to evaluate our approach with networks that feature a larger structure and to study the feedback of different groups of users. Especially, when dealing with a larger structure the calculation of the similarity matrix necessary for the hierarchical clustering we used can become very time consuming. Hence, the integration of more robust and sophisticated clustering methods may be beneficial for the analysis. In this regard, the adaptation of caching and querying strategies as introduced in [1] for an interactive clustering of time series according to different time intervals may also be beneficial for an interactive analysis of larger networks. Additionally, at the moment, we provide visual cues to communicate if changes to clustering settings may yield additional information. Here, an integration of additional visualization such as the *TreeJuxtaPoser* [22] that shows two trees side by side to facilitate the direct comparison of the clustering results may also support the analysis.

Furthermore, our approach focused on the abstraction of the structure according to the temporal development of attributes. Thus, we reduce the number of nodes and edges to be analyzed. An alternative would be to focus on the reduction of the number of time points to be visualized. A basis for such an approach could be the idea of clustering the time points according to the spatial configuration as presented in [2, 16] for the analysis of spatio-temporal data. In this way, only time points showing different structural configurations have to be analyzed.

Acknowledgments

The authors wish to thank Thomas Mundt for his support during the data gathering as well as Andreas Dähn and Hans-Jörg Schulz for their valuable feedback on earlier versions of this paper. This work was supported in part by the DFG graduate school MuSAMA. The Figures 5 and 8 use imagery © OpenStreetMap contributors licensed according to CC-by-SA (see `http://www.openstreetmap.org/copyright`).

References

[1] Z. Ahmed and C. Weaver. An adaptive parameter space-filling algorithm for highly interactive cluster exploration. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 13–22, 2012.

[2] G. Andrienko, N. Andrienko, S. Bremm, T. Schreck, T. Von Landesberger, P. Bak, and D. Keim. Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns. *Computer Graphics Forum*, 29(3):913–922, 2010.

[3] D. Archambault. Structural differences between two graphs through hierarchies. In *Proceedings of Graphics Interface*, pages 87–94, 2009.

[4] D. Archambault, H. Purchase, and B. Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, 2011.

[5] U. Brandes and S. R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. *Information Visualization*, 2(1):40–50, 2003.

[6] M. Burch and S. Diehl. TimeRadarTrees: Visualizing dynamic compound digraphs. *Computer Graphics Forum*, 27(3):823–830, 2008.

[7] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf. Parallel edge splatting for scalable dynamic graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2344–2353, 2011.

[8] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). RFC 3626, Internet Engineering Task Force, October 2003.

[9] S. Diehl and C. Görg. Graphs, they are changing. Dynamic graph drawing for a sequence of graphs. In *Proceedings of Graph Drawing*, pages 23–31, 2002.

[10] A. R. T. Donders, G. J. van der Heijden, T. Stijnen, and K. G. Moons. Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, 59(10):1087–1091, 2006.

[11] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 52–58, 2006.

[12] Y. Frishman and A. Tal. Dynamic drawing of clustered graphs. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 191–198, 2004.

[13] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):727–740, 2008.

[14] S. Hadlak, H.-J. Schulz, and H. Schumann. In situ exploration of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2334–2343, 2011.

[15] M. Harrower and C. Brewer. Colorbrewer.org: An online tool for selecting colour schemes formaps. *The Cartographic Journal*, 40(1):27–37, 2003.

[16] P. Köthur, M. Sips, J. Kuhlmann, and D. Dransch. Visualization of geospatial time series from environmental modeling output. In *EuroVis-Short Papers*, pages 115–119, 2012.

[17] J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, 1983.

[18] G. Kumar and M. Garland. Visual exploration of complex time-varying graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):805–812, 2006.

[19] G. Li, Y. Wang, L. Zhang, and X. Zhu. Similarity measure for time series based on piecewise linear approximation. In *International Conference on Wireless Communications Signal Processing*, pages 1–4, 2009.

[20] T. Lopatic. OLSRD link quality extensions. `http://www.olsr.org/docs/README-Link-Quality.html`, 2004. Last accessed on 2013-03-25.

[21] M. Luboschik, H.-J. Schulz, C. Maus, A. Uhrmacher, and H. Schumann. Heterogeneity-based guidance for exploring multiscale data in systems biology. In *Proceedings of the 2nd IEEE Symposium on Biological Data Visualization*, pages 33–40, 2012.

[22] T. Munzner, F. Guimbretiére, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics*, 22(3):453–462, 2003.

[23] M. Rohrschneider, A. Ullrich, A. Kerren, P. F. Stadler, and G. Scheuermann. Visual network analysis of dynamic metabolic pathways. In *Proceedings of the 6th international Conference on Advances in Visual Computing - Part I*, pages 316–327, 2010.

[24] P. Saraiya, P. Lee, and C. North. Visualization of graphs with associated timeseries data. In *IEEE Symposium on Information Visualization*, pages 225–232, 2005.

[25] L. Shi, C. Wang, and Z. Wen. Dynamic network visualization in 1.5D. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 179–186, 2011.

[26] A. Telea. Combining extended table lens and treemap techniques for visualizing tabular data. In *Proceedings of the Eighth Joint Eurographics / IEEE VGTC conference on Visualization*, pages 51–58, 2006.

[27] Y. Tu and H.-W. Shen. Visualizing changes of hierarchical data using treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1286–1293, 2007.

[28] A. Unger and H. Schumann. Visual support for the understanding of simulation processes. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 57–64, 2009.

[29] J. J. van Wijk and E. R. van Selow. Cluster and calendar based visualization of time series data. In *Proc. of IEEE InfoVis'99*, pages 4–9, 1999.

[30] T. Warren Liao. Clustering of time series data-A survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[31] T. Wollenberg. Performance measurement study in a wireless OLSR-ETX mesh network. In *Wireless Days (WD), 2012 IFIP*, pages 1–7. IEEE, 2012.

[32] H.-M. Wu, S. Tzeng, and C.-H. Chen. Matrix visualization. In C.-H. Chen, W. Härdle, and A. Unwin, editors, *Handbook of Data Visualization*, pages 681–708. Springer, 2008.

[33] J. S. Yi, N. Elmqvist, and S. Lee. TimeMatrix: Analyzing temporal social networks using interactive matrix-based visualizations. *International Journal of Human-Computer Interaction*, 26(11 & 12):1031–1051, 2010.

[34] H. Ziegler, M. Jenny, T. Gruse, and D. Keim. Visual market sector analysis for financial time series data. In *IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 83–90, 2010.