# Visualizing Internet Evolution on the Autonomous Systems Level*

Krists Boitmanis**, Ulrik Brandes, and Christian Pich

Department of Computer & Information Science, University of Konstanz

**Abstract.** We propose a visualization approach for large dynamic graph structures with high degree variation and low diameter. In particular, we reduce visual complexity by multiple modes of representation in a single-level visualization rather than abstractions of lower levels of detail. This is useful for non-interactive display and eases dynamic layout, which we address in the online scenario.
Our approach is illustrated on a family of large networks featuring all of the above structural characteristics, the physical Internet on the autonomous systems level over time.

## 1 Introduction

Visualization of large evolving relational data sets is a challenging task, because the size of the data and dynamics are difficult to deal with even in isolation. A visualization problem that encompasses these features simultaneously is the macroscopic view of the evolving Internet topology on the autonomous-systems (AS) level. To the best of our knowledge, there are no dynamic visualization approaches that can produce purely structure-based drawings of a sequence of AS graphs in reasonable time.

In this paper we propose to attack this problem by first applying a few complexity reduction operations, which lead to both considerably smaller graphs and savings of screen space. However, instead of hiding the less important parts of a graph, which is a common approach to reduce complexity, we still show them in the drawing with different representation modes. The reduced graphs are laid out with a stress majorization approach [14] enhanced with a novel scheme for calculating distances between nodes that is specially suited for graphs with extremely skew degree distributions. Also, the flexibility of the stress majorization technique allows to adapt it for the dynamic setting. This is demonstrated in the online scenario, where the previous drawing is respected during the layout for the next time point.

The paper is structured as follows. In Sect. 2, we give a brief review of the AS-level Internet topology and related work. The layout method for static

* Research partially supported by DFG under grant Br 2158/3-1 (ECRP). Part of this research was done while the first author was visiting the Institute of Mathematics and Computer Science, University of Latvia.

** Corresponding author: Krists.Boitmanis@uni-konstanz.de

snapshots of the graph and our complexity reduction operations are the subject of Sect. 3 and the extension of this approach to dynamic graph visualization and its application to AS graphs are presented in Sect. 4. Section 5 concludes the paper with a short discussion.

## 2 AS-Level Internet Topology and Related Work

An *autonomous system*, or *AS* for short, is a group of computer networks typically under the same administrative authority, using the same routing policy. The Internet can thus be analyzed in terms of connections and interactions between ASes. The *AS graph* is then a model for the Internet, having ASes as nodes and AS-to-AS connections as edges.

In recent years, analysis of the AS-level Internet topology has attracted interest of many researchers. The common goal is to keep track of structure and dynamics of the Internet, to develop meaningful and robust models explaining such observations, and to come to reasonable interpretations. Technically and economically, the analysis has manifold practical aspects, e.g. for improving reliability, routing efficiency, and fairness.

Interest in the AS graph excelled when power-laws and scale-free distributions were observed to be characteristic features [12]. Since then, various aspects of autonomous systems have been investigated, such as inferring AS graphs from collected data [15], modeling and generating artificial AS graphs [16], and comparison of measured and generated data [23], to name just a few examples. The dynamics of the AS graph are analyzed in [13]; models for the AS graph evolution and a comparison of AS graph inference methods from different data sources are given in [18].

Visualization and visual analysis of AS graphs have been attempted as well, though to a lesser extent. Probably best known are the circular drawings from the Skitter project of CAIDA [9]. HERMES [7] is a system for orthogonal drawings of the Internet hierarchy or parts thereof. Force-directed generation of Internet maps is the approach taken in the Internet Mapping Project [8]. The two-and-a-half dimensional drawings of AS graphs in [3] are based on a hierarchy of increasingly denser cores, which is also used in [2]. Dynamics in the routing behavior of autonomous systems are visualized by LinkRank [17], animations for network performance assessment are described in [6]. To the best of our knowledge, only the layouts in [3] consider the complete AS graph and are purely structure-based.

A number of approaches for drawing general dynamic graphs have been proposed [5], but few principles and frameworks are prevalent [4, 10, 11].

As a test ground for the methods we developed, we have constructed AS graphs at various time points from the BGP (Border Gateway Protocol) route data available in the archives of the *Route Views* project [21]. The structure of each AS graph is inferred from a collection of AS paths consisting of a sequence of numbers. Two ASes are connected by an undirected edge if their numbers appear consecutively in at least one of the AS paths.

## 3 Static Layout and Complexity Reduction

Although our ultimate goal is to visualize a sequence of AS graphs, we first restrict ourselves to visualizing a single snapshot $G = (V, E)$.

### 3.1 Layout Method

We have chosen the stress majorization approach as the graph layout method [14]. This choice was motivated by the quality of the resulting drawings, the flexibility of the approach facilitating adaptations for the dynamic setting, existing speed-up techniques, and simplicity of implementation at least when the localized stress minimization is used. Note, however, that other methods with similar properties, e.g. variants of force-directed methods, could be used equally well.

The basic idea is an iterative minimization of the stress function

$$\mathrm{stress}(X) = \sum w_{uv}(\|X_u - X_v\| - d_{uv})^2 \ , \tag{1}$$

where the sum extends over all unordered pairs of nodes $\{u, v\}$ in $V$. Here $X_v \in \mathbb{R}^2$ is the position of the node $v \in V$, $d_{uv}$ is the ideal distance between the nodes $u$ and $v$, which is usually the length of a shortest path in $G$, and $w_{uv}$ is a non-negative weight allowing different pairs of nodes influence the stress measure differently. Weights $w_{uv} = d_{uv}^{-2}$ are a common choice.

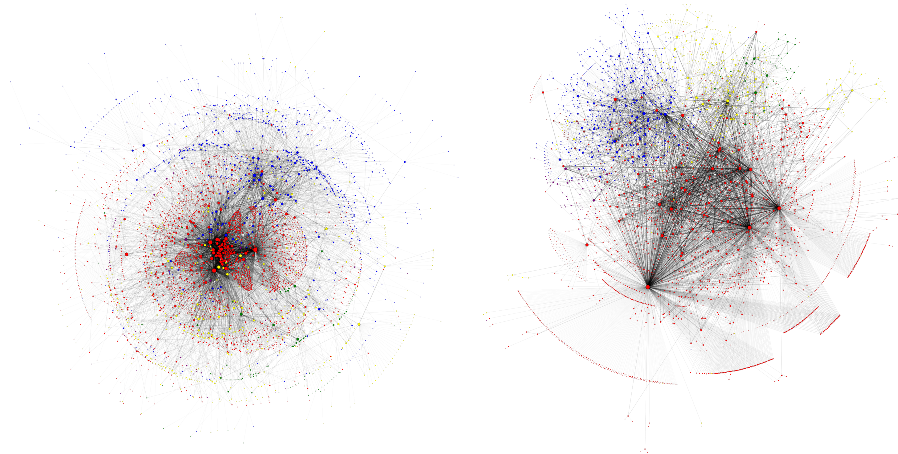
We can confirm the claim that the above strategy "makes the neighborhood of high degree nodes too dense" [14] unless appropriate lengths are assigned to edges (Fig. 1(a)). This is due to the extremely skewed degree distribution of AS graphs; the AS graph in Fig. 1 has 4271 nodes, 75% of which have a degree one or two, while a few extreme nodes have degrees as large as 924, 673, and 470. The problem is somewhat remedied if the geometric mean $\sqrt{d_u d_v}$ of the degrees of nodes $u$ and $v$ is used as the length of an edge $e = \{u, v\} \in E$, because then the high-degree nodes strive to push their neighbors further away (Fig. 1(b)). In Sect. 3.3 we propose a novel method for calculating distances that further improves the quality of drawings.

We use the following graphical conventions throughout the paper.

- The area of a node is proportional to the squared logarithm of its degree.
- The opacity of an edge is proportional to the radius of its smaller end-node. In effect, edges between high-degree nodes attract more attention of an observer.
- The nodes are colored according to the continents the corresponding ASes belong to: we use blue to represent Europe, red for North America, yellow for Asia, purple for South America, brown for Africa, and green for Oceania.

### 3.2 Visual Complexity Reduction

This section presents our attempts to allay the visual clutter of drawings by using different representation modes without loosing any information.



(a) (b)

**Fig. 1.** A snapshot of the AS graph in the year 1998 – (a) uniform edge length, (b) degree-dependent edge length.

First, consider the typical AS graph in Fig. 1 with its many nodes of degree one. In a standard representation, these result in large fans that form dominant visual features that consume large areas but represent the least interesting structures. To remove this effect, we use radial clustergrams [1, 20], a compact representations of trees, as follows:

- Let $T \subset V$ be the set of nodes in the attached trees of $G$, which can be obtained by an iterative removal of the leaves of $G$ until all remaining nodes have degrees two or more.
- Draw the induced graph $G[V \setminus T]$ in the standard representation with nodes as circles and edges as straight lines.
- Draw the nodes of $T$ as radial clustergrams around the nodes in $V \setminus T$ they are attached to.

Our radial clustergrams are slightly different from those in [1, 20] to maintain the degree-area correlation. Suppose that the children $v_1, v_2, \ldots, v_k$ of a node $v$ have to be drawn inside an annulus wedge with the radius $r$ and the angle $\alpha$ (Fig. 2(a)). The desired area $S_i$ of each node $v_i$ is fixed because it is derived from its degree. Moreover, we require that the radial width $w$ of the children of the same node is equal. Clearly, $w$ cannot be less than $w_{\min} = \sqrt{\frac{2}{\alpha}\sum_{i=1}^{k} S_i + r^2} - r$. On the other hand, we would also like to avoid very thin nodes, so $l_i/w \leq c$ must hold for some constant $c > 0$, where $l_i$ is the length of the outer arc of $v_i$. A possible solution to this inequality is given by the largest root $w_i$ of the cubic equation $cw^3 + 2crw^2 - 2S_i w - 2S_i r = 0$, and consequently the common layer width for all children of $v$ is calculated as $w = \max\{w_{\min}, w_1, w_2, \ldots, w_k\}$. Note, that the annulus wedge is not filled completely if $w > w_{\min}$ (Fig. 2(b,c)).

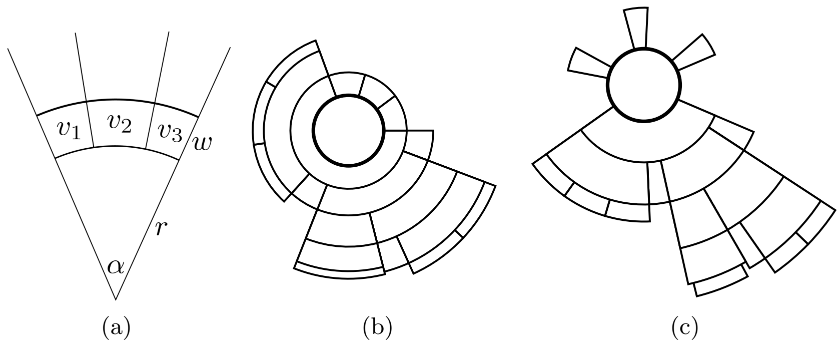



**Fig. 2.** (a) Children of the same node drawn in a specified annulus wedge. (b) A radial clustergram without restrictions on the radial width of nodes. (c) A radial clustergram of the same tree when the radial width of nodes is bounded from below.

Figure 3 shows a layout of the AS graph with the attached trees drawn as radial clustergrams. Although the clutter is somewhat reduced, there are still plenty of low-degree nodes around the periphery and many of them seem to be connected to the same set of core nodes. The latter is a structural feature that we emphasize by aggregating the equivalent nodes as follows.

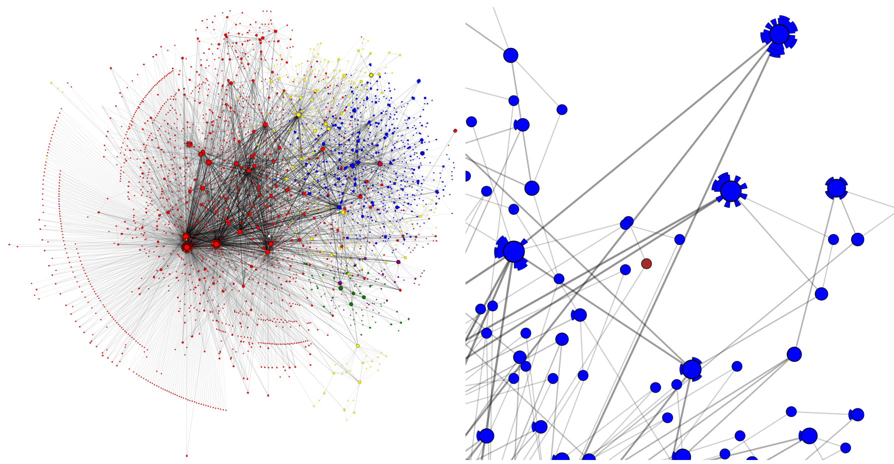
- Construct the equivalence classes of the relation $\{(u,v)|u,v \in V \setminus (T \cup N(T)) \wedge N(u) = N(v)\}$. Note that nodes with attached trees are considered as special and not equivalent to anything else.
- Contract each non-trivial equivalence class $U \subseteq V$ of this relation into a new meta-node $v_U$ before applying the layout.
- After the position of a meta-node $v_U$ has been determined by the layout algorithm, restore the equivalent nodes $U$ and draw them around the position of $v_U$ in a compact way. A good choice is the sunflower placement from [22, 19].

As can be seen in Fig. 4(a), some sets of equivalent nodes are quite large and the compact placement shows their neighbors much better.

The final complexity reduction step consists of replacing maximal induced paths $(v_0, v_1, \ldots, v_k)$ by direct edges $\{v_0, v_k\}$ between their ends, provided that the inner nodes $v_i$ $(0 < i < k)$ are not affected by the previous two reductions, i.e. $v_i \notin T \cup N(T) \cup M$, where $M$ is the set of meta-nodes. After the layout of the reduced graph is calculated, the induced paths are restored and drawn straight between their ends (in the rare cases when two or more paths run between the same pair of end nodes, these paths are drawn parallel without mutual overlaps).

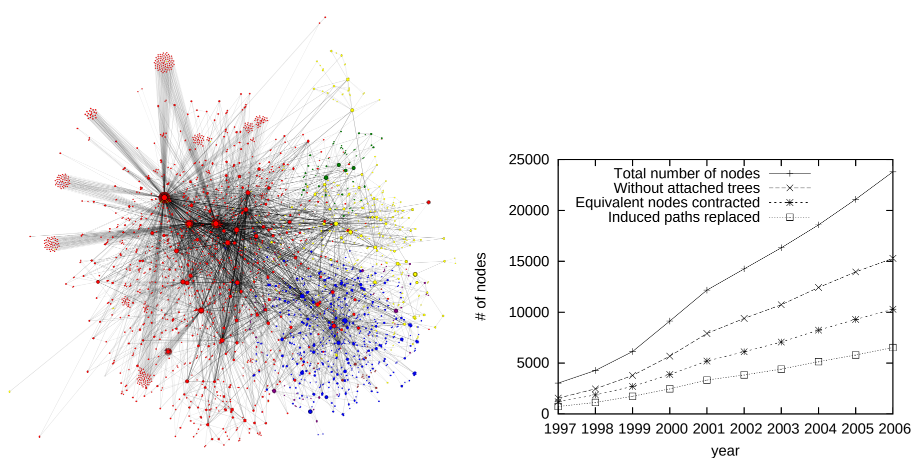
A side effect of these reduction operations is a lower number of nodes, which is a very significant advantage as the full stress majorization considers the distances between every pair of nodes. Figure 4(b) shows the growth of the AS graph over a decade and how many nodes remain after each reduction step.

In what follows, we assume that the graphs are reduced according to these three operations.



(a) (b)

**Fig. 3.** Full (a) and zoomed-in (b) drawings of the AS graph in the year 1998 with attached trees drawn as radial clustergrams.




(a) (b)

**Fig. 4.** (a) The same AS graph after further complexity reductions. (b) The effect of the reduction operations on the number of nodes.

### 3.3 Layout Method – Revisited

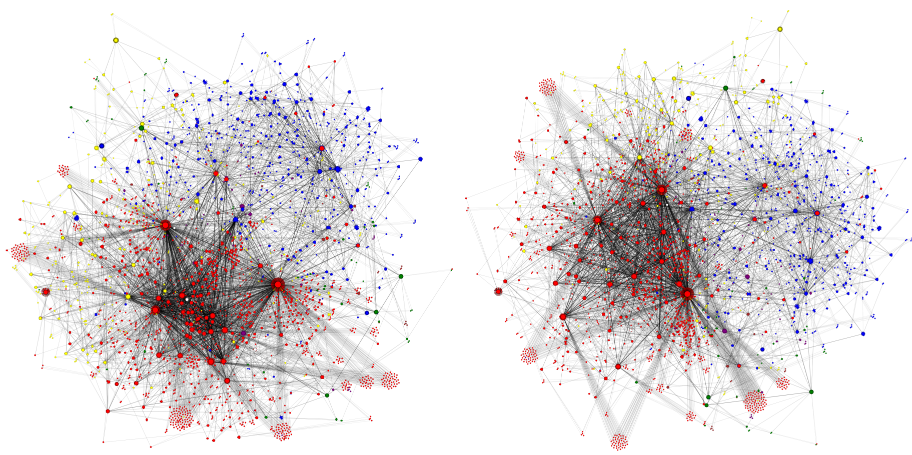
The drawing in Fig. 4(a) leaves something to desire in terms of quality. First, the high-degree nodes are still placed too close to each other obscuring the structure of how they relate to the rest of the graph. Secondly, some low-degree nodes with only high-degree neighbors end up as peaks on the periphery because the length of their incident edges is unnecessarily high. A novel approach for calculating the pairwise distances and their weights solves both of these problems (Fig. 5(a)).

*Edge Lengths.* The importance of an edge $e = \{u, v\} \in E$ is captured better if its length $l_e$ is an increasing function of the smallest degree $\min\{d_u, d_v\}$ of its ends. In our experiments the best results were obtained with $l_e = \ln(\min\{d_u, d_v\})$. In this way, adjacent nodes of high-degree are placed far apart and their connecting edge is more prominent. On the other hand, the incident edges of low-degree nodes are drawn much shorter so that these nodes are placed close to their neighbors.

*Distances.* Special care must be taken when calculating pairwise distances from these re-scaled edge lengths. We cannot simply use shortest paths in the weighted graph $G$, because two high-degree nodes are still very close if they have a common neighbor of low degree. Distances are therefore calculated as $d_{uv} = \max\{l(P) | P \in \mathrm{SUP}(u, v)\}$, where $\mathrm{SUP}(u, v)$ denotes the set of shortest paths between $u$ and $v$ in the unweighted graph $G'$ underlying $G$ and $l(P)$ is the length of the path $P$ in the weighted graph $G$. In other words, we consider a longest weighted path among those with a minimum number of edges. Such distances can be easily calculated in $O(|V||E|)$ time by performing a breadth-first-search from each node $v \in V$ and determining the longest weighted paths in the shortest paths dag with source $v$. Also, the unweighted distances $d_{G'}(u, v)$ should be used when calculating the weights in (1), i.e. $w_{uv} = d_{G'}(u, v)^{-2}$, because otherwise the important distances would be outweighed by less important ones. An exception to this rule are the meta-nodes representing groups of equivalent nodes. If two meta-nodes $u$ and $v$ have a common neighbor, we use $w_{uv} = 1$ rather than $1/4$ to make it less likely that the resulting sunflowers would overlap. Moreover, the “degree” of a meta-node $v_U$ representing a set $U$ of equivalent nodes is assumed to be $\sum_{v \in U} d_v$ such that it represents the total “importance” of all nodes in $U$.

*Speed-Up.* The final modification of the method concerns its running time. It took 25 minutes to create a drawing of an AS graph having 23,779 nodes and 49,706 edges on a computer with 2 GHz CPU and 2 GB of memory, which is largely due to the use of the full distance matrix. Fortunately, the method can be sped up without affecting layout quality considerably (compare the two drawings in Fig. 5). The idea is to calculate the layout in two phases. First, a small subset of nodes $P \subseteq V$ with the highest degrees is chosen as pivots (we used 200 pivots in our experiments), and these are laid out in the above technique according to the distances $d_{uv}$, $u, v \in P$. In order to position the nodes in $V \setminus P$, we again utilize stress majorization, but fix pivots and ignore all distances $d_{uv}$, $u, v \notin P$

unless $\{u, v\} \in E$. In this way, we ignore a very large number of "inessential" distances, and the running time drops from 25 minutes to 44 seconds. It should be noted that this approach is slightly different from the sparse stress approach of [14], although they are similar in that the overall structure of the drawing is determined by some important core nodes, and other nodes are laid out based on distances to those core nodes and nodes in some close neighborhood. The main difference lies in the two applications of the stress majorization, which leads to the pivots being placed independently from the rest of the graph. This two-phase technique turned out to be more successful in our setting.



(a) (b)

**Fig. 5.** Drawings of the same AS graph obtained by the full stress majorization using the modified distances (a) and the fast two-phase method (b).

## 4 Dynamic Layout

In this section we will modify the above method to be applicable to dynamic graphs in the online scenario, i.e. when an existing drawing of the graph is respected during the creation of a subsequent drawing.

Suppose that besides the graph $G = (V, E)$ we are given the desired positions $p_v \in \mathbb{R}^2$ for nodes $v$ in a subset $U \subseteq V$, which are the result of a preceding layout. In order to preserve the overall view of the evolving graph, we have an additional criterion now to minimize the distance of nodes from their desired positions. Following the ideas in [4], we can do this with the stress majorization technique in a rather straightforward way by augmenting the stress with node displacement penalties, $\text{stress}(X) = \text{stress}_{\text{quality}}(X) + \text{stress}_{\text{stability}}(X)$, where $\text{stress}_{\text{quality}}(X)$ is defined as in (1) and $\text{stress}_{\text{stability}}(X) = \sum_{v \in U} w_{\text{st}} \|X_v - p_v\|^2$. The stability parameter $w_{\text{st}}$ can be adjusted to trade the quality of the drawing for the stability. Figure 6 shows how the value of the quality stress function increases and the total movement of nodes decreases when the stability parameter increases.

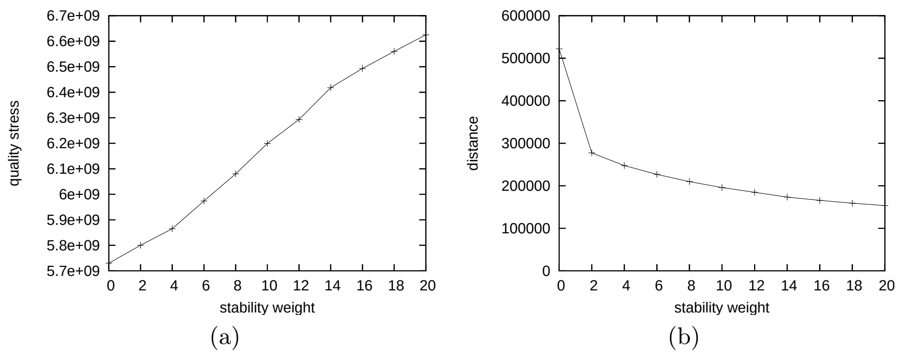



**Fig. 6.** The effect of the stability parameter on the quality of the drawing (a) and the total movement of nodes (b) when the online method is applied to the AS graph in the year 1998. The desired positions are obtained from the layout of the graph at the year before.

Figures 7 and 8 show a selection of the resulting drawings when the fast two-phase stress majorization is applied in the dynamic online scenario for annual snapshots of the AS graph from 1997 to 2006.[1] A stability of $w_{\text{st}} = 20$ was used for creating these drawings.

## 5 Conclusion

We combined loss-less complexity reduction operations with tailored stress majorization techniques to produce drawings of a large evolving graph with skewed degree distribution, specifically the Internet on the level of autonomous systems. Even though the density of AS graphs increases rapidly over time, we believe that such a macroscopic view of the Internet can reveal evolution patterns, possibly supported by additional information coded in graphical attributes. It would be very interesting to see if our visualizations can actually help monitoring the evolving Internet.

## References


1. D.K. Agrafiotis, D. Bandyopadhyay, and M. Farnum. Radial clustergrams: Visualizing the aggregate properties of hierarchical clusters. *Journal of Chemical Information and Modeling*, 47(1):69–75, 2007.


[1] The full animated sequence can be downloaded from `http://www.inf.uni-konstanz.de/algo/research/asgraph/`.

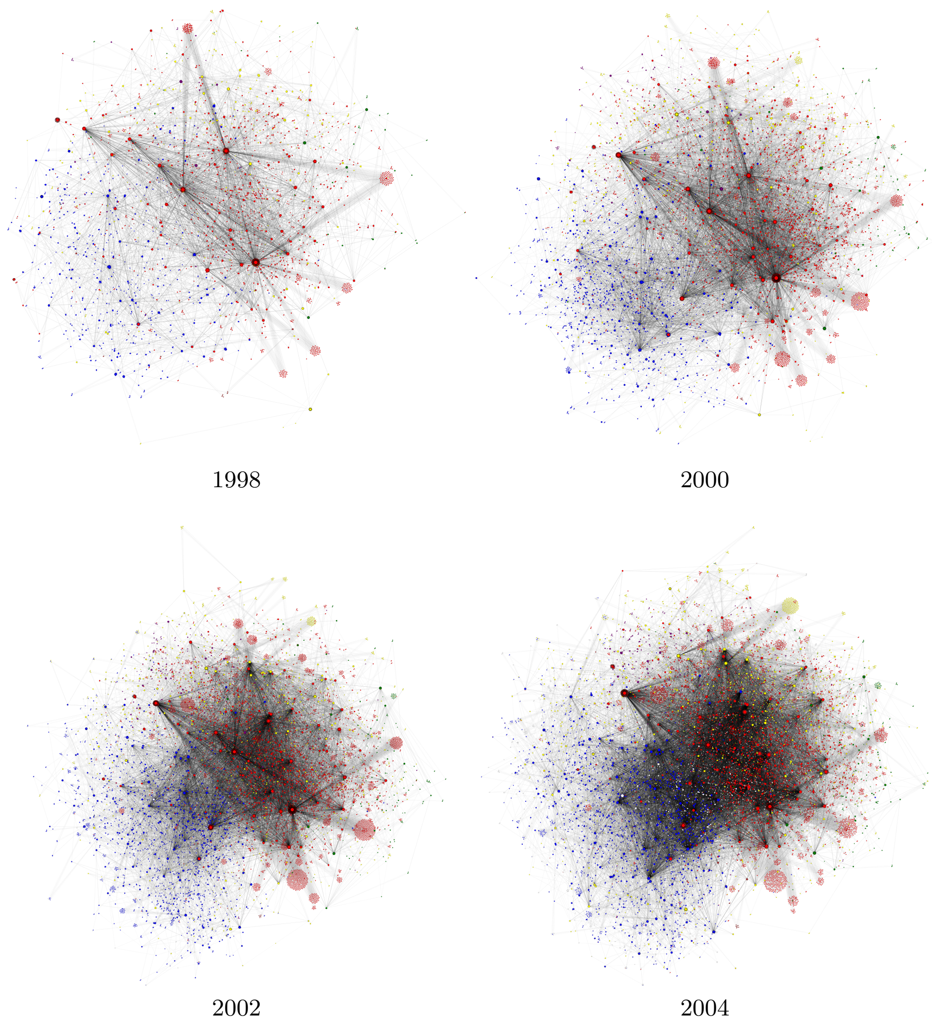



**Fig. 7.** Drawings of the evolving AS graph obtained from dynamic stress majorization in the online scenario

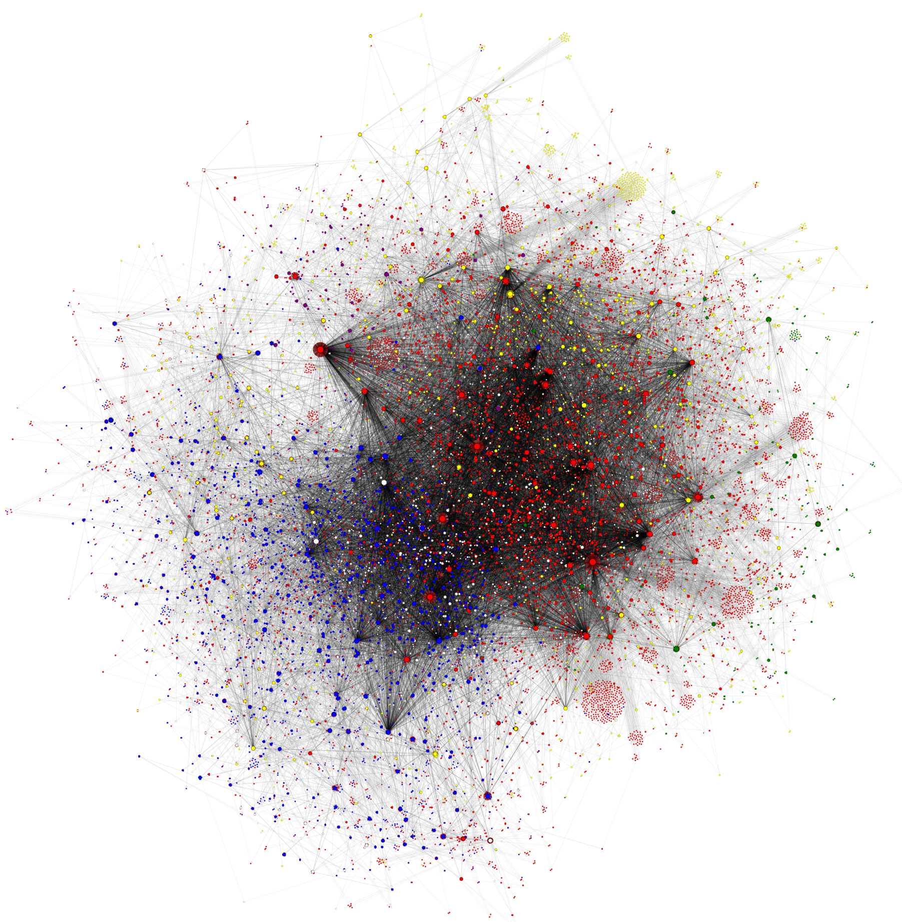


**Fig. 8.** Drawing of the 2006 AS graph finishing the sequence of Fig. 7

2. J.I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In *Advances in Neural Information Processing Systems 18*, pages 41–50, 2006.
3. M. Baur, U. Brandes, M. Gaertler, and D. Wagner. Drawing the AS graph in 2.5 dimensions. In *Proc. Graph Drawing*, pages 43–48, 2004.
4. U. Brandes and D. Wagner. A Bayesian paradigm for dynamic graph layout. In *Proc. Graph Drawing*, pages 236–247, 1997.
5. J. Branke. Dynamic graph drawing. In M. Kaufmann and D. Wagner, editors, *Drawing Graphs*, pages 228–246. Springer LNCS 2025, 2001.
6. J. Brown and A. McGregor. Network performance visualization: Insight through animation. In *Proc. Passive and Active Measurement Workshop*, pages 33–41, 2000.
7. A. Carmignani, G. Di Battista, W. Didimo, F. Matera, and M. Pizzonia. Visualization of the autonomous systems interconnections with hermes. In *Proc. Graph Drawing*, pages 150–163, 2000.
8. B. Cheswick, H. Burch, and S. Branigan. Mapping and visualizing the internet. In *Proc. USENIX Annual Technical Conference*, pages 1–12, 2000.
9. Cooperative Association for Internet Data Analysis (CAIDA). Visualizing Internet topology at a macroscopic scale. `http://www.caida.org/analysis/topology/as_core_network/`.
10. S. Diehl and C. Görg. Graphs, they are changing – dynamic graph drawing for a sequence of graphs. In *Proc. Graph Drawing*, pages 23–30, 2002.
11. C. Erten, P.J. Harding, S.G. Kobourov, K. Wampler, and G. Yee. Graphael: Graph animations with evolving layouts. In *Proc. Graph Drawing*, pages 98–110, 2004.
12. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proc. ACM SIGCOMM*, pages 251–262, 1999.
13. M. Gaertler and M. Patrignani. Dynamic analysis of the autonomous system graph. In *Proceedings Inter-Domain Performance and Simulation*, pages 13–24, 2004.
14. E.R. Gansner, Y. Koren, and S. North. Graph drawing by stress majorization. In *Proc. Graph Drawing*, pages 239–250, 2004. Full version available at `http://www.research.att.com/~yehuda/pubs/majorization_full.pdf`.
15. L. Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Transactions on Networking*, 9(6):733–745, 2001.
16. C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. Technical Report CSE-TR-433-00, University of Michigan, 2000.
17. M. Lad, D. Massey, and L. Zhang. Visualizing internet routing dynamics using link-rank. Technical report, UCLA, 2005.
18. P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K.C. Claffy, and A. Vahdat. The internet as-level topology: Three data sources and one definitive metric. *ACM SIGCOMM Computer Communication Review*, 36(1):17–26, 2006.
19. P. Neumann, M. Sheelagh, T. Carpendale, and A. Agarawala. Phyllotrees: Phyllotactic patterns for tree layout. In *Proc. EuroVis*, pages 59–66, 2006.
20. J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald. An evaluation of space-filling information visualizations for depicting hierarchical structures. *Int. J. Hum.-Comput. Stud.*, 53(5):663–694, 2000.
21. University of Oregon. Route views project. `http://www.routeviews.org/`.
22. H. Vogel. A better way to construct the sunflower head. *Mathematical Biosciences*, 44:179–189, 1979.
23. S. Zhou and R.J. Mondragón. Redundancy and robustness of AS-level internet topology and its models. *Electronics Letters*, 40(2):151–152, 2004.