The moəpTi*k*Z package

a library for typesetting computer networks

Stephan M. Günther

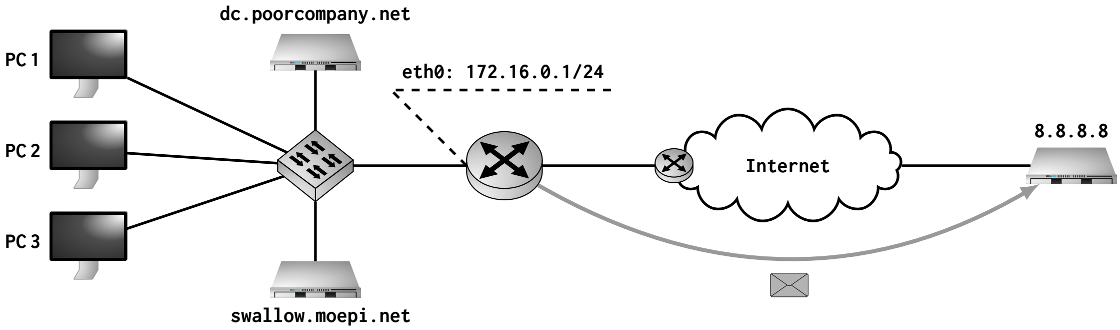
July 23, 2015

The moəpTi*k*Z package provides shapes for typesetting computer networks, e. g. shapes for hubs, switches, routers, and more. It is in use since 2012 for the lecture materials of our basic course in computer networks due to shortcomings of some commercial products and (at that time) the lack of meaningful alternatives. For the first official release many shapes were redesigned and a couple of new features added, e. g. complete anchor support and custom colors. Currently the package it provides the following shapes and commands:

- `hub`, `switch`, `router`, `client`, `server`, `nuc`, `messageclosed`, `messageopen`
- `\tikzextractx`, `\tikzextracty`

1 What it can do

Did you ever try to build slides for a networking lecture from scratch? If so, do you care about off-topic details such as using solely vector graphics, perfect alignment of nodes, easy modification of hundreds of figures when you decide to change fonts or colors, and doing all that without even thinking about a pointing device? Then you probably like what moəpTi*k*Z may do for you:



2 The shapes

Figure 1 presents the different shapes as printed by `\node[<shape>] at ...` (the label font is typewriter here, but that's up to you). The default label shows the name of the corresponding shape. The gray box around the shape is a unit square making the shapes bounding box visible. The for red crosses mark the label positions specified by a node option such as `label=<pos>:<labeltext>`. The gray bullets demonstrate automatically calculated anchor positions.

If the node option `minimum size` is omitted, nodes will be drawn in a bounding box of 1 cm × 1 cm. Since not all shapes fill their bounding box, default label positions may vary between nodes, e. g. label positions of server and client shapes differ. This may be undesirable proved to be better than having node labels too far away from the visible shape.

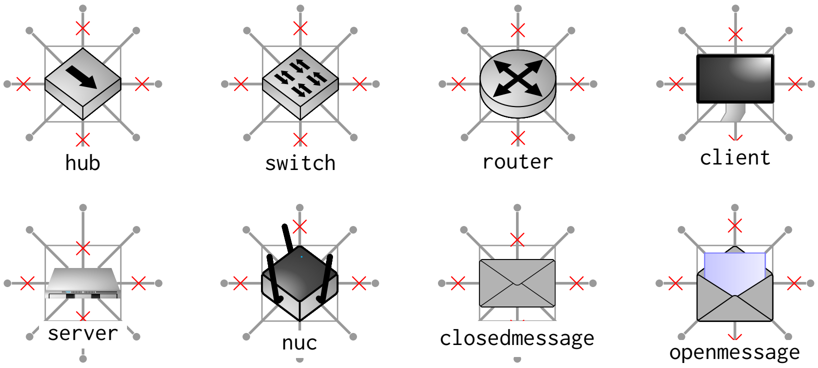
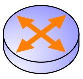


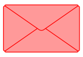
Figure 1: Shapes currently defined

2.1 Changing node colors

There are very few occasions where you might want to highlight a specific node by changing its color. Or you are upset by the default mark color, i e., the color of arrows on top of some shapes. You may change it at will: all shapes support custom colors using the `fill=<color>` option. In addition to that, the message shapes also allow to change the line color using the `draw=<color>` option:



```
\node[router,fill=blue,draw=orange] at (0,0) {};
```



```
\node[messageclosed,fill=red!40,draw=red] at (4,0) {};
```

```
\node[messageopen,fill=orange!40,draw=orange] at (8,0) {};
```

2.2 Upcoming features

There will be an option to disable shadings, i. e. the option will disable the “lighting effects" of shapes. Of course, there will be some additional shapes such as `firewall` and `workstation` soon. Feature requests are welcome. And of course there will be many code updates, which is all but nice pgf right now.

3 The commands

moəpTi*k*Z provides two commands to make your live with easier:

1. `\tikzextractx{(x,y)}{\<macroname>}`
   Extracts the `x`-component of a coordinate and stores it `\<macroname>` for later use.
2. `\tikzextracty{(x,y)}{\<macroname>}` Extracts the `y`-component of a coordinate and stores it `\<macroname>` for later use.