Flexible Traffic Matrix Analyser for Inter-domain Network Operation and Planning

Andreas Kock,
T-Systems Nova GmbH Berkom,
Berlin, Germany

Abstract

A traffic matrix is an abstraction of traffic exchanges measurement results between nodes in an network. It is needed by ISPs for a wide variety of traffic engineering tasks. Each of these tasks has its special kind of input. All of these inputs, which can be related to a network structure, can be subsumed under the concept of traffic matrix. In this paper we introduce a flexible traffic matrix analysing model. We derive it from a traffic matrix systemization and the user's needs, and proof its completeness.

1 Introduction

Traffic matrix calculation is of major interest to ISPs for traffic engineering including forecasting future customer demands on bandwidth as well as finding bottlenecks in the current network state or revealing network abuses such as DoS-attacks, traffic characteristics in contrary to the ISP-customer contract, etc. Calculating traffic matrices of any type is a costly task as it involves either direct monitoring of all the traffic or QoS parameters of interest for a given pair source/destination , or coupling partial measurements with inference techniques. Both ways include calculating a table with a size of squared order in the number of origins and destinations to be observed. That is why usually there are calculated only those parts of the matrix, which belong to the sources and/or destinations of special interest.

2 Taxonomy of traffic matrices

2.1 Systemization

To systematize the types of traffic matrices [1] introduced a two-level taxonomy of traffic matrices based on the spatial representation of network traffic used and the aggregation level for the sources and destinations engaged in traffic exchanges. The spatial representation level contains point-to-multipoint, point-to-point, and path-point-to-point traffic matrices. The aggregation level includes POP level, router level, link level, and prefix level.

In detail:

> A **point-to-point traffic matrix** represents a traffic demand between a pair of source and destination nodes.
>
> A **path-point-to- point traffic matrix** represents a traffic demand between a pair of source and destination nodes using a dedicated path.
>
> A **point-to-multipoint traffic matrix** represents a traffic demand between a single source and a set of destination nodes.

2.2 Calculating traffic matrices

Traffic matrices can be constructed on two different ways – direct measurement, and estimation.

Direct measurement typically involves collecting packet-level traces or flow-level traces and coupling them with routing information. Alternatively, one can use monitoring equipment, that collects statistics on packets and/or flows and reports the processed byte counts. This approach results in a number for each source/destination link, which is highly accurate with the monitoring equipment as the main source of error.
Estimating traffic matrices on the other hand uses predefined information such as traffic demands, previously measured demand distributions, etc. to calculate a number, a function, function parameters or a set of these, that characterize the relation (that is, the traffic volume, etc.) for a certain period of time.

2.3 Purpose of traffic matrices

Calculating traffic matrices is not a job that is standalone. Traffic matrices can be used for various tasks, such as capacity planning and dimensioning, routing, operations and management, customer reports, and supporting service level agreements. Each of these tasks involves deriving special values from at least a small subset of matrix elements. Additionally, each task has special demands on the contents of the traffic matrix elements.

3 Design of traffic matrix analyser

3.1 Reflection

Although traffic matrices are such an important source of information, it is difficult to find the appropriate one for a given application as well as it is expensive creating them by using traffic data. Since the amount of data incurring when evaluating traffic matrices is practically not manageable for a human user, it is necessary to prepare them suitably, in order to ease future processing. Special items for this are an appropriate visualisation as well as problem-oriented support, which means coupling traffic matrices with distance measures and geographical, political or organisational classification of origin and destination of a traffic flow.

3.2 Summarizing the demands to construct a traffic matrix analyser

As stated in section 2 constructing traffic matrices depends on the following categories:

1. **Organisational**

   The traffic matrix analyser must be able to deal with spatial representation using point-to-point, point-to-multipoint, and path-point-to-point information as well as aggregation representation at POP level, router level, link level, or prefix level.

2. **Mathematical**

   The traffic matrix analyser must be able to deal with single numbers as well as functions and sets of all of it, to support multiple flows over a link.

3. **Purpose of the traffic matrix**

   The traffic matrix analyser must be able to support any task to be done using it. Therefore the matrix content should be freely usable as well as it must be possible to include special processing applications.

3.3 Conclusion

From the theory of graphs and networks it is well-known, that a graph consists of nodes and edges. A set of consecutive edges is called a path.

Therefore, from the **organisational** point of view, we need the structure elements

1. **MatrixNode**

   representing a network node in any matrix type except point-to-multipoint

2. **MultiMatrixNode**

   representing a set of nodes in a point-to-multipoint matrix

3. **MatrixLink**

   representing a link in a point-to-point matrix as well as a relation between a MatrixNode and a MultiMatrixNode in a point-to-multipoint situation

4. **MatrixPath**

   representing a path in a path-point-to-multipoint situation

Additionally, an information is needed describing the aggregation level of a node, that is POP, router, link, or prefix level. This can be managed using an attribute to the MatrixNode.

From the **mathematical** point of view the objects above are extended the following ways:

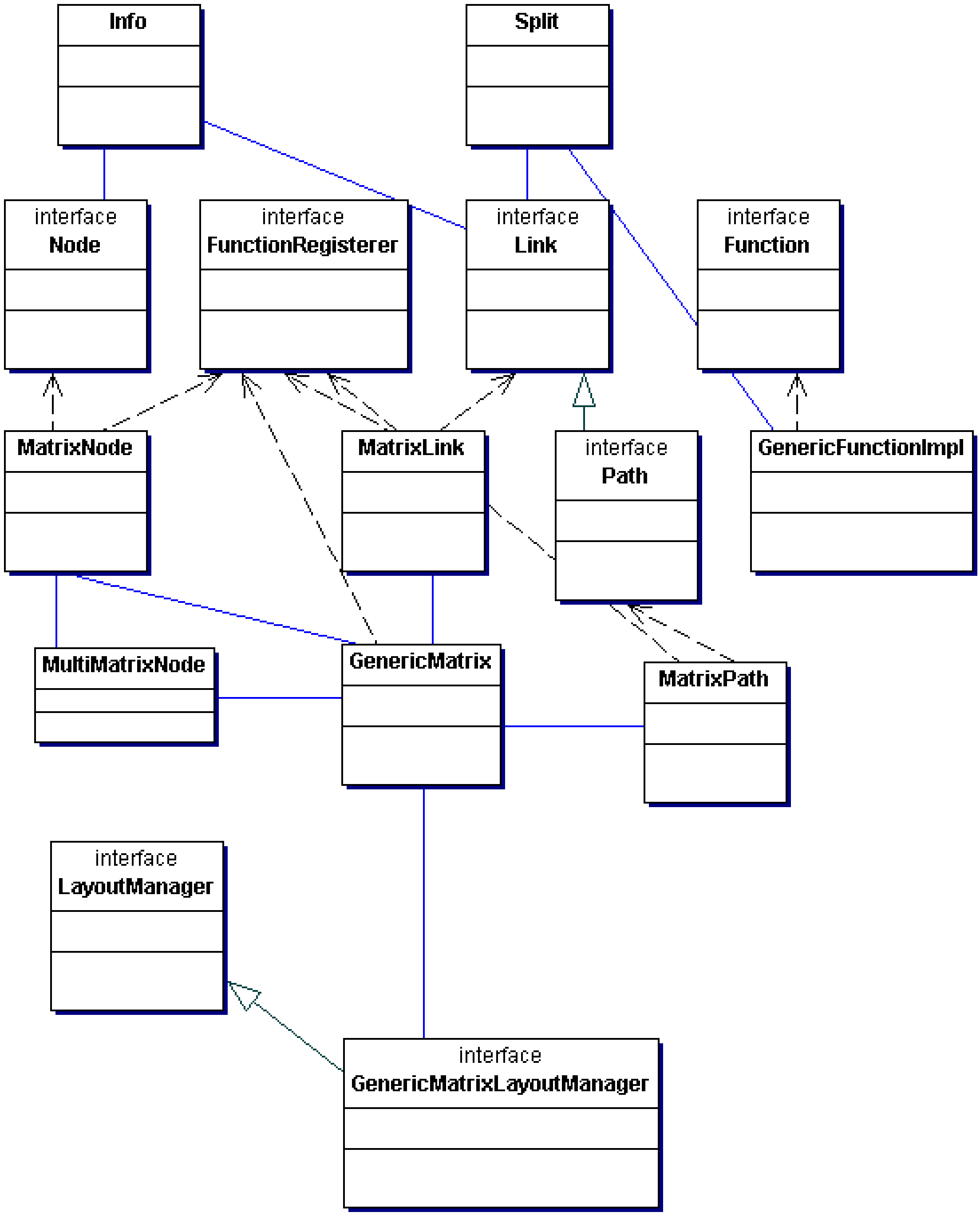
A network link (and a path as well) can be used by more than one flow simultaneously. This can be represented by a split object added to the MatrixLink (MatrixPath) that holds the values for each flow. A flow value itself is represented by a function returning any object of users choice, such as a constant, a function, etc.

From the point of view of **matrix use**

The purpose of the traffic matrix is to support network evaluation (traffic engineering, security, etc.). For each of these tasks, procedures are specified working on MatrixNodes, MatrixLinks, MatrixPaths, and MultiMatrixNodes. To couple these procedures with the matrix elements we register each procedure at the appropriate matrix element.

The last purpose – visualisation – can be decoupled from the matrix using the concepts of layout and layout managers. Each layout manager takes the matrix and lays it out using its own rules.

This all results in the following class diagram:



4 Proof of completeness

The above stated class structure is complete. This is shown in the following section by comparing it with the demands from section 3.2.

1. **Organisational**

   a) Point-to-point matrices are constructed using
   * MatrixNodes
   * MatrixLinks

   b) Path-Point-to-point matrices are constructed using
   * MatrixNodes
   * MatrixPaths

   c) Point-to-Multipoint matrices are constructed using
   * MatrixNodes
   * MultiMatrixNodes
   * MatrixLinks

   d) The aggregation level is represented by an additional Info-object to the MatrixNode
   or by an additional attribute.

2. **Mathematical**

   a) Numbers for characterizing a link are stated using a function that returns a constant.

   b) Functions for characterizing a link are stated using the function returning an object
   of choice.

   c) Sets for characterizing a link with multiple flows are stated using splits that contain
   a function for each link.

3. **Purpose of the traffic matrix**

   a) Special methods over a matrix are coupled with it or its components.

   b) Visualization is done using layout managers.

5 Converting traffic matrices using the abstract model

Recall, that we have two traffic matrix information levels, i.e. spatial representation and aggregation level. For the spatial representation level [Medina et al.] stated, that the only possible conversion at POP, router, or link level is from path-point-to-point to point-to-point-matrix.

Path-point-to-point matrices are constructed using the objects MatrixNode and MatrixPath.

PPP={(s,d,p,V(s,d,p)) |
s,d – MatrixNode,
p – MatrixPath from s to d,
V - Volume of traffic from s to d via p}

Point-to-point matrices are constructed using the objects MatrixNode.

PP={(s,d,V(s,d)) |
s,d – MatrixNode,
V-Volume of traffic from s to d}

Mapping from PPP to PP is then

$$M: PPP(s,d,p,V(s,d,p)) \Rightarrow PP(s,d,V(s,d))$$

$$s_{PPP} \Rightarrow s_{PP}$$

$$d_{PPP} \Rightarrow d_{PP}$$

$$V(s,d)PP=\sum_{\forall p(s,d)} V(s,d,p)_{PPP}$$

Source and destination node object are kept and MatrixPath objects are replaced by a MatrixLink object where the new split is the sum of all splits of the paths.

In the prefix aggregation level it is stated, that point-to-multipoint and point-to-point matrices are identical. That gives us a trivial mapping between these two types of matrices. The second possible conversion, i.e. path-point-to-point to point-to-point, works the same way as stated above, and therefore converting path-point-to-point to point-to-multipoint becomes trivial, too.

For the aggregation level, traffic matrix conversion is much easier as it only involves mapping from finer level of aggregation to higher level of aggregation. For the model it is:

1. Compute the mapping tables
   Prefix-to-Link
   Link-to-Router
   Router-to-POP

2. Replace the attribute to the MatrixNode, if applicable, using these mapping tables.

3. Sum the splits, if applicable.

6 Populating the traffic matrix

Populating the traffic matrices is one of the interesting questions. As the traffic analyser is intended to be universal, the input data should reflect the internal data structure as well as the analyser should be as lenient as possible accepting every input that looks like traffic matrix data. Following, an XML-Schema is given showing the structure of such an input:

```
<xsd:schema
xmlns:xsd="http://www.w3.org/1999/XMLSchema
">

<xsd:complexType name="Attribute">
  <xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
    <xsd:attribute name="name" type="xsd:string"/>
    <xsd:attribute name="value" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="AttributeList">
  <xsd:element name="Properties"
        type="Attribute"
            minOccurs="0"
            maxOccurs="*"/>
</xsd:complexType>

<xsd:complexType name="NodeType">
<xsd:sequence>
  <xsd:attribute name="id" type="xsd:ID"/>
  <xsd:element name="NodeProperties"
        type="AttributeList"
            minOccurs="0"
        maxOccurs="1"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="MatrixNodeList">
  <xsd:element name="Node"
        type="NodeType"
        minOccurs="1"
            maxOccurs="*"/>
</xsd:complexType>

<xsd:complexType name="ShareType">
  <xsd:sequence>
    <xsd:Attribute name="id" type="xsd:ID"/>
    <xsd:Attribute name="name"
type="xsd:string"/>
    <xsd:Element name="Function">
      <xsd:Attribute name="name"
type="xsd:string"/>
      <xsd:Attribute name="parameter"
type="xsd:string"/>
    </xsd:Element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SplitType">
  <xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
    <xsd:element name="Share"
          type="ShareType"
                minOccurs="1"
          maxOccurs="*"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SplitListType">
    <xsd:element name="Split"
          type="SplitType"
          minOccurs="0"
          maxOccurs="*"/>
</xsd:complexType>

<xsd:complexType name="LinkType">
  <xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
    <xsd:attribute name="origin"
type="xsd:IDREF"/>
    <xsd:attribute name="destination"
type="xsd:IDREF"/>
    <xsd:element name="LinkProperties"
          type="AttributeList"
          minOccurs="0"
          maxOccurs="1"/>
    <xsd:element name="SplitList"
          type="SplitListType"
          minOccurs="0"
          maxOccurs="1"/>
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="LinkListType">
  <xsd:element name="Link"
         type="LinkType"
         minOccurs="0"
         maxOccurs="*"/>
</xsd:complexType>

<xsd:complexType name="TMDocument">
  <xsd:sequence>
    <xsd:element name="Origins"
          type="MatrixNodeList"/>
```

```
    <xsd:element name="Destinations"
                 type="MatrixNodeList"/>
    <xsd:element name="LinkList"
                 type="LinkListType"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

7 Summary

In this paper, a flexible traffic matrix analysing model is proposed, based on an 2-level traffic matrix classification, taken from a proposal by [1]. The advantage of the model is, that it allows constructing a generic framework for using traffic matrices.
This traffic matrix model is complete for this 2-level matrix classification, and allows easy traffic matrix conversions, if applicable.

8 References

[1] Medina, et al. A Taxonomy of IP Traffic Matrices.

[2] Fabio Arciniegas. XML Developer’s Guide. Franzis. 2001.