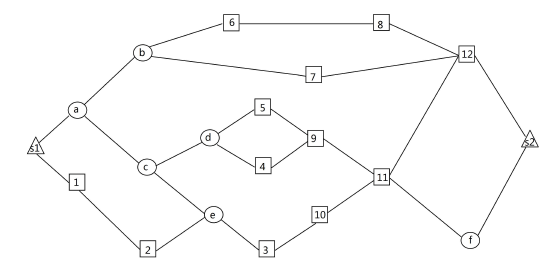
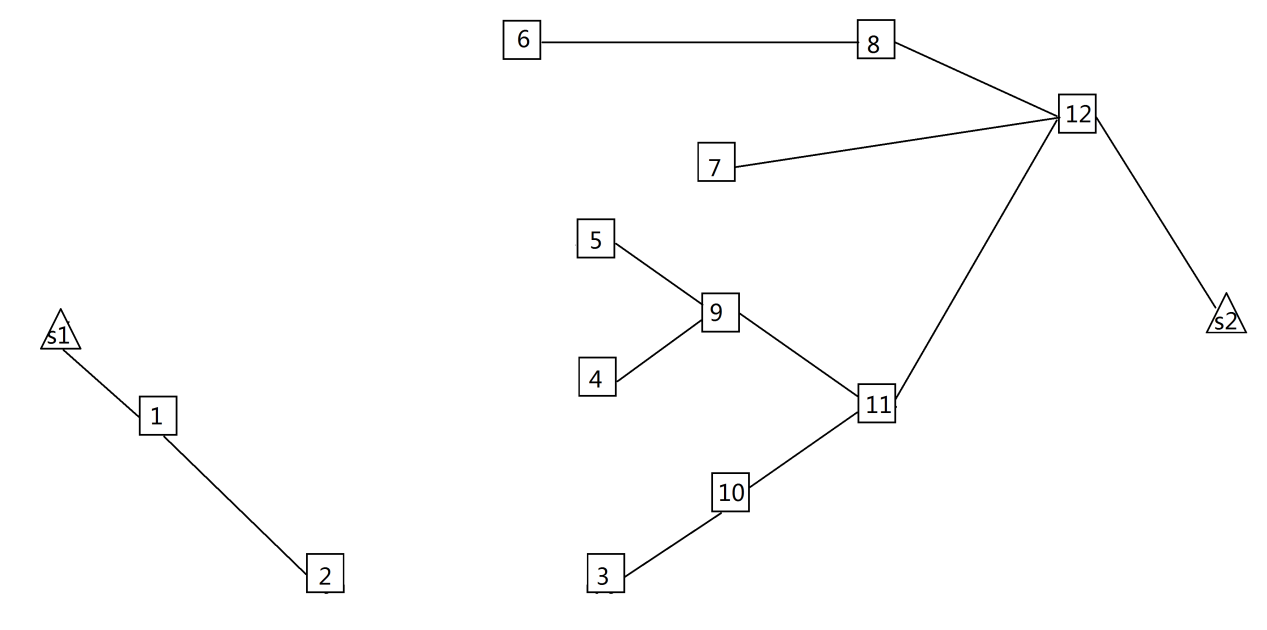
# Introduction

Ontology modeling is a useful tool to solve some historic problem. My major is about computer network routing. When people get or distribute information through internet, data flow must be forwarded to different destination. Network device such as router and switch is one important part of network; we can abstract them as a node in network, as the following figure.



Square nodes denote as destinations; triangle nodes denote as sources; circle nodes denote as common nodes. Our application scenario is that all destinations must connect source node and the links they need is as less as possible.

The problem is to seek a routing as follows.



So data can be transmitted according to the routing, and in order to achieve better effect, we wish the links of routing is minimum.

Unluckily, network request is changeable. For example, we usually wish data not to enter into certain a device which maybe is a secrecy network. Hence, it’s hard to model a ontology. Related work about ontology routing is rare.

We usually calculate the routing through special algorithm. But it repeatedly calculate many times and the advantage of ontology is the procedure of learning. After network company settles down network device and constructs a local network, the structure of network is relatively fixed and almost doesn’t change. It is suitable for a structure-fixed network to build ontology model. Then, with routing request increasing, network ontology is able to directly apply the request without calculating the requests.

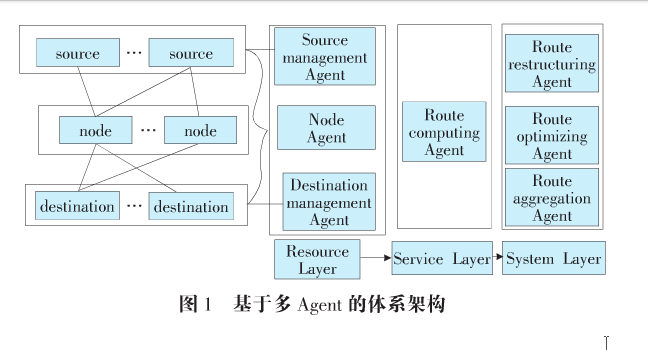
Actually, data from source to destination is transmitted along a sequence of network nodes. Our routing is composed of these nodes and edges, denoting as links between devices.

# Related work

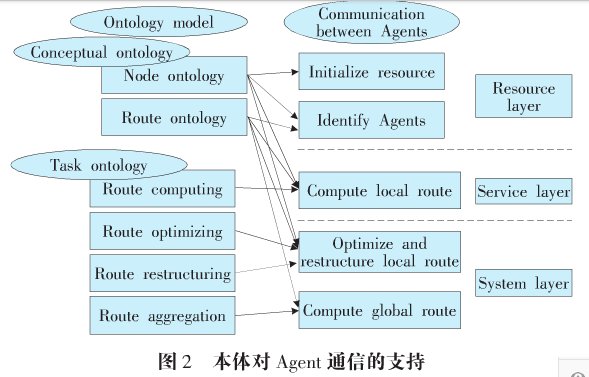
The last section explain what I want network ontology to do. Actually, in this field many people and scholar also endeavor to route data based on ontology model.

I investigate two paper about network routing.

The first paper is for traditional network routing problem. But it is designed as a complicated framework as follows.



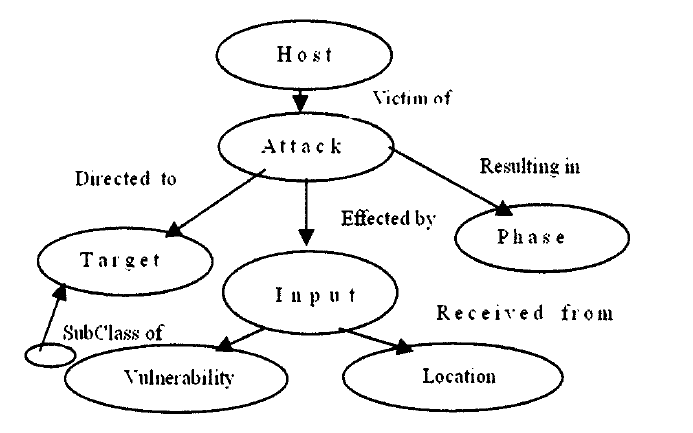
The author use agents to calculate routing based on ontology.



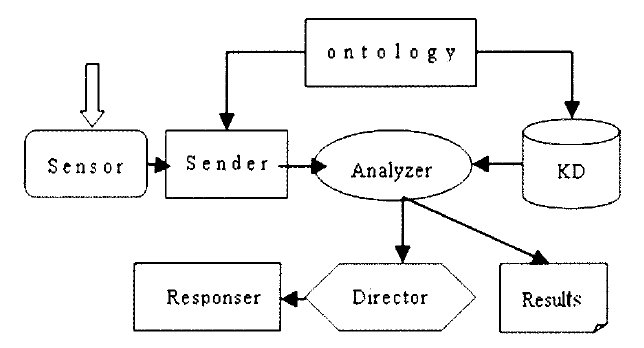
This paper is short but inspires me to construct a routing ontology. We concentrate other ontology in on

The second paper is about network security.

Security is a totally different problem. Its ontology is like follows.



Its author uses this ontology like follows.



# Intuition

My ontology is for multicast routing problem – to get routing from network. The innovation is to simplify topology ontology and redefine axioms for our special application scenario. We need connect all destination nodes to source nodes using paths. Paths consist of links.

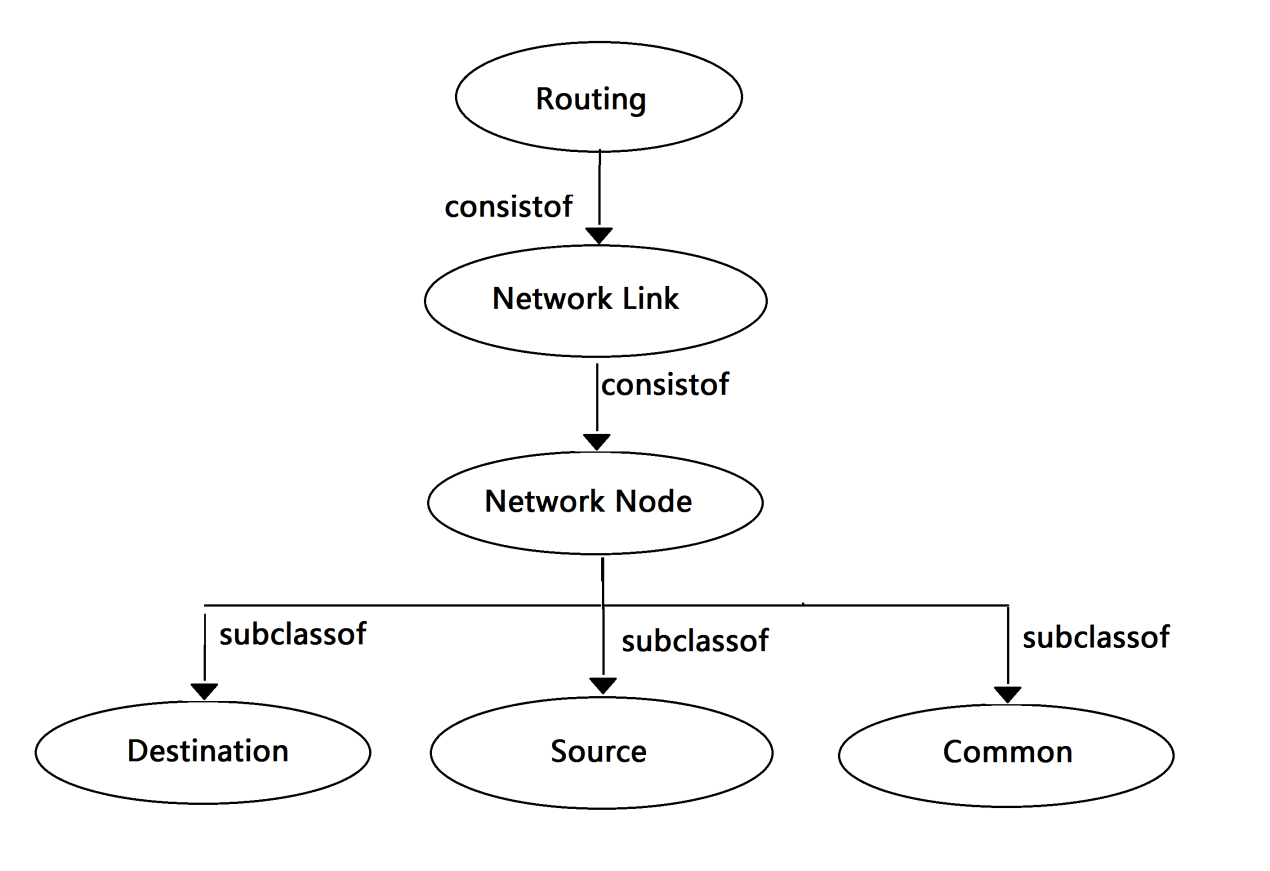
At first, our ontology should ensure routing is normal. That is to say,

1. no isolated nodes exist;
2. one link connects only two different nodes;
3. routing must include at least two nodes. One is destination and the other is source.

Then, we wish to get better performance even though routing connect destinations to sources. The less links in routing is, the better the routing is. Hence, we make some constraints to enhance the routing’s performance. These constraints are:

1. Not all sources must be included in routing, but at least one source exists.
2. All destinations must be included in routing. Hence, one link from each destination to its neighbor exists in routing.
3. Every node except source in routing is connected to one source.
4. Common nodes have at least two neighbors.

# Terms and concept



The basic concepts are routing. Routing is a set of links. These link may be connected, either disconnected. One link may be a routing when source is single one node and destination is single one node.

If two nodes in routing is connected, the path exists. The unit of a path is a link. One link is determined if we know its two end nodes. And we call the relationship of two nodes is neighbor.

Source, Destination and Common are three types of Nodes. One node must belong to one and only one type of Nodes. Actually, one simple routing must include destination and source.

### Predicate

Unary predicate *Routing(a)* denotes as an instance of routing.

Unary predicate *Path(a)* denotes as an instance of path.

Unary predicate *Link(a)* denotes as an instance of link.

Unary predicate *Node(a)* denotes as an instance of node.

Unary predicate *Destination(a)* denotes as an instance of destination.

Unary predicate *Source(a)* denotes as an instance of source.

Unary predicate *Common(a)* denotes as an instance of common.

Unary predicate *Connect(a,b)* indicates any two source is connected.

Binary predicate *Neighbour(a,i)* denotes as a neighbor node instance i of node a.

Binary predicate *ConsistofLink (a,i)* denotes as an end node instance i of link a.

Binary predicate *ConsistofRouting(a,i)* denotes as an link instance i of routing a.

# Axioms

### General axioms

1. no isolated nodes exist;



This axiom requires, for each node *a* there always exists one link *l* and *l* connects *a*. Hence, *a*  is not isolated.

1. one link connects only two different nodes;



Given that *i* and *j* are two different end nodes of one link *k*, if *l* is also one end node of *k*, *l* is either *i* or *j*. Hence, each link includes only two different end nodes.

1. all nodes are disjoint.



If node *x* is destination, *x* is not source or common.

If node *x* is source, *x* is not destination or common.

If node *x* is common, *x* is not source or destination.

### Advanced axioms

1. Axioms for routing.



If two routings have the same set of links, they are the same routing.

1. Axioms for neighbor node.



If two different nodes *i* and *j* are connected through link *l*, we call they are neighbor nodes.

1. Axioms for destination node.



For each destination, there always exists one neighbor and they are connected through one link. Hence, destinations are connected in routing.

1. Axioms for link.



This axiom requires, if two links have the same two end nodes, they are the same link.

1. Axioms for source



Routing includes at least one source. Hence, at least one source isn’t isolated and there always exists one link connecting the source.

1. Every node except source in routing is connected to one and only one source.



For each node *x*, if *x* isn’t source, there always exists one source a connecting *x*.

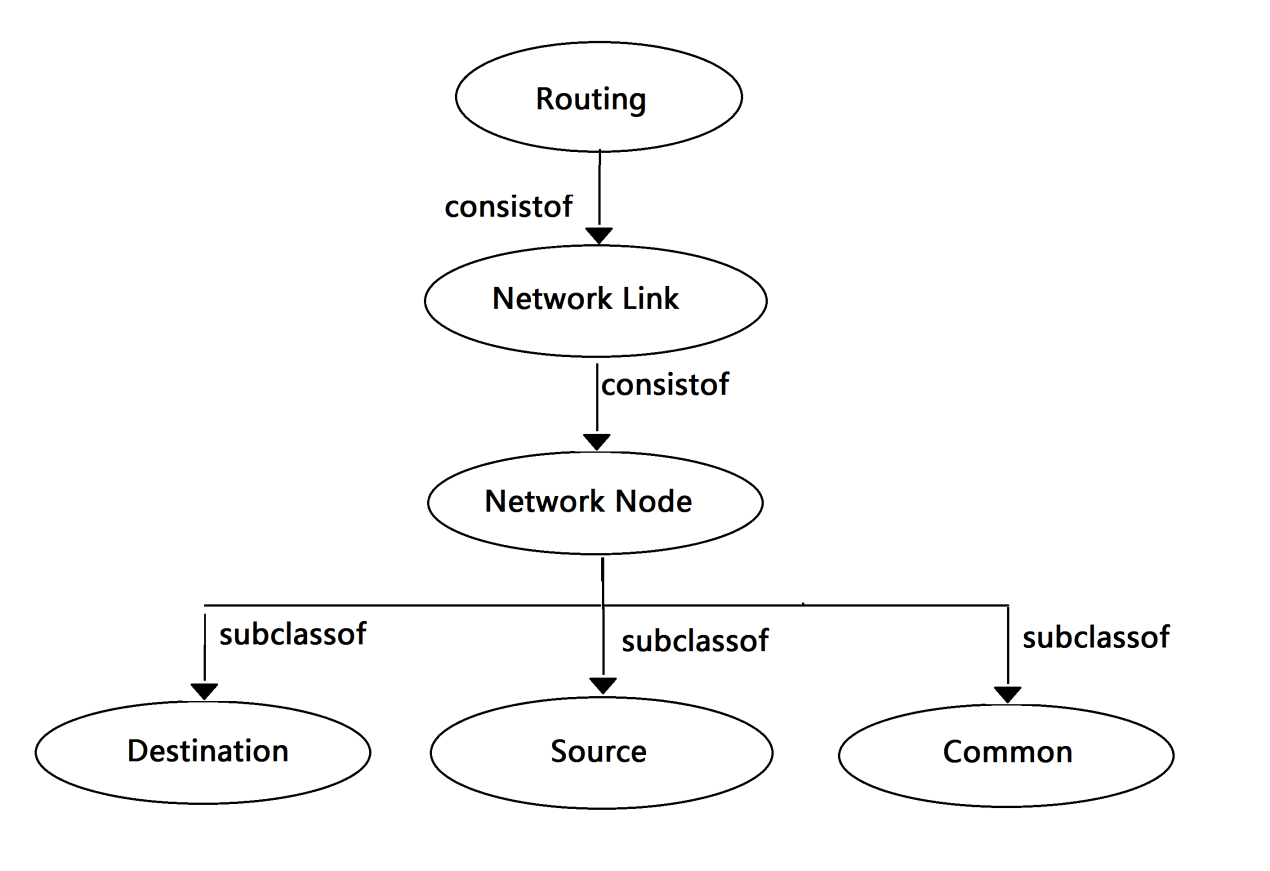
But *x* isn’t connected to any other sources *b*.

1. Axioms for common node



This axiom requires that, each common node has two different neighbor nodes. If one common node has three or more neighbors, it still conforms to this axiom.

# Model specification



Let denote as the structure of routing.

1. Property 1, element of is :



denotes as all  should conform to  , such that



elements of  will be referred to as link.

1. Property 2, element of is :



denotes as all  should conform to  , such that



elements of  will be referred to as node.

1. Property 3, element of is divided into three parts:



such that



elements of  will be referred to as common, elements of  will be referred to as destination, elements of  will be referred to as source .

1. Model for link

The model is based on the following axiom,



each link includes two different nodes.

The other axiom is advanced axiom 4. It requires each link connects two and only two nodes.



1. Model for routing



1. Model for node

Elements of node are divided into three types,



Such that,



Elements of C are referred to as common nodes, elements of D are referred to as destination nodes, elements of S are referred to as source nodes.

1. Model for common node

We require each common node must be connected to two different neighbor nodes.



1. Model for other nodes

For each other node, if they aren’t source.

We require each of them must be connected to one source.



# Consistency proof

### General axioms

1. Axiom 1 requires each node in routing must be not isolated.



iff for any node x and link z , if 

then x isn’t belong to routing because property 5 indicates any routing is consist of links. x isn’t belong to any link, and hence isolated node x is also belong to any routing.

1. Axiom 2 requires any link connects two and only two different nodes.



Iff for any two different nodes i and j, if there always exists node l and



then according to property 4, one of following holds:



1. Axiom 3 indicates that each node belong to one and only one type,



iff for any , one of the following holds:



Which holds by property 6.

### Advanced axioms

1. Every routing is determined by its set of link.



iff for any routing x, y, if

,

then

 ,

which holds by property 5.

1. Axiom 2 indicates the definition of neighbor node.



iff for two different nodes i and j, if one link k satisfies



then the following holds



which holds by property 4.

1. Axiom 3 requires all destinations must be connected to neighbor. This axiom indicates that all destinations are included in routing.



iff for any connected destination i, there always exists one neighbor j satisfying



then there always exists one link l including i



hence according to property 5, routing always includes the set of links,



and the destination i is also included in routing.

1. Axiom 4 indicates that one link is determined by its two end nodes.



Iff for any two links l and k, there always exist two pair nodes satisfying



which holds by property 4. If



then



1. Axioms for source

At first, at least one source is included in routing,



iff there always exists one source s and its neighbour i, if



then there exists one link j satisfying



According to property 5, routing consists of the set f links and



Hence,



1. Axiom 6 requires all nodes except source must be connected to one and only one source.



iff for any one node *x,* if



then there always exists one node *a* satisfying



which holds by property 8.

1. Axiom 7 requires any common node have at least two different neighbor.



iff any one common node *i*, according to property 8, if



then its two different neighbor nodes *j* and *l* satisfying



which holds by property 8.