DA233X Degree Project in Computer Science and Engineering, specializing in Machine Learning

Polyxeni Ioannidou - ppio@kth.se

July 14, 2020

Planning and detecting anomalies in computer networks

1 Background

1.1 Automated network configurations - Planning in routing protocol

1.1.1 Problem description

Automated planning is a part of Artificial Intelligence, where the execution of a sequence of actions results in the achievement of the goal for a given problem. More precisely, an informed search algorithm uses a heuristic evaluation function to find the desired solution, which must be optimized. In some cases, it is guaranteed that it can find the best possible solution. In addition to the property of optimality, in AI planning a search algorithm has the property of completeness, which means that if a solution exists, then it is guaranteed that the search algorithm will find it.

Based on these properties of optimality and completeness, automated planning can probably produce a dynamic network configuration that outperforms static configurations in a computer network. In other words, the need for automatically configuring or reconfiguring a large interconnected computer network leads to develop a program that controls all activities, tasks, changes, inputs, and features in the network under certain constrains. As a result, using the advantages that automated planning offers, we can probably improve the already existing static configurations. Although, we have to underline that the existing static configurations in many cases perform pretty well.

In CERN's network, the routing dynamic protocol used is the Open Shortest Path First (OSPF). In a network, OSPF considers that all routers are indiscernible from each other, as all of them execute the same routing algorithm in order to find a routing path and deliver their data in the most efficient way. They are organized in Autonomous Systems (AS), where each AS is a group of devices that runs the same routing algorithm and share information about each other. The OSPF routing is used as the routing algorithm that runs in the intra-autonomous system. Therefore, it is also called an intra-autonomous system routing algorithm. All in all, our purpose is to investigate the advantages of a dynamic protocol (e.g. OSPF) over a static one. The main difference is that a dynamic protocol benefits from the search algorithm that is applied, which motivates us to compare and contrast different search algorithms in order to derive our conclusion.

1.1.2 What is a routing protocol

Before we analyze the OSPF routing protocol in the context of planning, we ought to describe what a routing protocol is and why OSPF is used, or in other words, what makes it a good choice. To start with, a routing protocol, or else a routing policy, is a set of rules or procedures that routing algorithms use to determine optimal package transfer and communication paths between the nodes in a network. Routing protocols arrange the router communication and the network topology understanding.

The purpose of the routing algorithm is to detect "good paths" in a network of devices. In particular, a "good path" is a sequence of devices from "sender" to "receiver" with the minimum cost. The definition of "cost" here may reflect the physical length of the corresponding link between two routers, or the link transmission time, or the link delay, or the link latency. In our case, it represents the link delay, which should be minimized in order to optimize the time of package transmission. In other words, the minimum the link delay, the faster a package gets delivered to a destination device. Therefore, the purpose of a routing algorithm is to detect the least costly paths between sources and destinations, given that a cost is assigned to every link in the network. There are several ways of discriminating between routing algorithms, e.g. static and dynamic, load-sensitive and load-insensitive, as well as centralized and decentralized routing algorithms. More specifically, a dynamic routing algorithm changes the paths when the traffic or the topology changes, whereas in a static routing algorithm routes do not change at all or they can be changed very slowly due to a human intervention. In a load-sensitive algorithm, the high costs are related with the congestion of a link, so a link's cost reflects its current level of congestion. For the last type of the routing algorithms, in a centralized routing algorithm every node has complete information about the costs of all network links, whereas in a decentralized algorithm no node has complete information about the costs of all network links [9, Chapter 5].

All in all, the purpose of routing protocols is to detect the network topology, create routing tables, and make routing decisions. The utility of these routing tables is that they have all the necessary data, with which one router chooses the best of its adjacency routers that will receive the delivered package and will transmit it to its best neighbor. Undoubtedly, the detection of the neighbors is decisive, since we aim to send a package only to those neighbors who can contribute to the minimization of the cost in the package transmission. Regarding all these, the routing protocols have three specific functions [1]:

- Learn and advertise routing information from and to neighboring routers.
- Obtain a map of all routes of the network that facilitates the topology determination.
- Calculate the best route, as well as the costs of every other route in the network. Then update all the available devices about any change that has been detected. The routers collect and store this information in routing databases.

Routing protocols can be categorized into three different types: distance-vector protocols, link-state protocols, and hybrid protocols. Nevertheless, this has nothing to do with the different types of routing algorithms that have been mentioned above. Regarding the distance-vector protocols, all the routers share the routing table, which is also updated when a device gets deactivated. Therefore, when every router broadcasts its routing information from all active interfaces, this increases the time that all devices have an accurate view of the network. We must underline that the routers here do not perform any mechanism so as to detect their neighbors because they can easily learn about that by the broadcasts they receive. With reference to the link-state protocols, they advertise routing information only with neighbors, but only when updates or changes

have been detected in the network. On the contrary, the distance-vector protocols broadcast the whole routing table periodically even if no changes have been detected. As a result, the usage of the bandwidth is more effective, which also leads to a faster convergence. In other words, link state updates are being sent when it is necessary to advertise that something has changed in the network, and then all the destination routers will reply with an acknowledgment. Lastly, the hybrid routing protocols can be described as a combination of the other two types. Hybrid protocols act as link-state protocols, i.e. they send updates only when they occur in the network, but they are built like the distance-vector protocols.

The question now is why is it important to know these details about the different types of routing protocols. The answer is that it can give us a good insight of why OSPF is such a good choice and why it has been used in such a large network as CERN's computer network.

1.1.3 Brief description of OSPF and explanation of its usage

OSPF is a link-state routing protocol, which means that it uses the whole network capacity in the most effective way. More precisely, with OSPF all devices in the network are getting updates when they occur, meaning that redundant broadcasts are omitted, and that the reachability information from all active interfaces is not shared periodically as in distance-vector protocols. Due to the fact that in OSPF a router/or device shares its routing information only with the direct neighbors, the need of discovering the neighbors is decisive. Therefore, the routers use the "hello" protocol, which could be likened to a handshake between two humans. The "hello" is a message that one router sends to all its interfaces and waits until it receives back a response to detect the active neighbors. Hence, this "hello" protocol increases the effectiveness of OSPF as a link-state protocol because once routers become neighbors with those active devices that are directly connected with them, they create adjacencies by exchanging link state databases. The databases are three different tables: one for the directly attached neighbors, one for the topology of the whole network, and one used as the routing table. Finally, the selection of the best route for each destination of the network is based on an algorithm called the Shortest Path First (SPF) algorithm.

In order to put into context the OSPF protocol, we should compare it with another protocol that is also used in many networks. We ought to compare OSPF with IS-IS, because they were both proposed to substitute the Routing Internet Protocol (RIP), although IS-IS was originally implemented for routing Open Systems Interconnection (OSI) traffic. In particular, OSPF was designed for the Transmission Control Protocol/Internet Protocol (TCP/IP) suite, while IS-IS was designed for the International Organization for Standardization (ISO) suite. The comparisons between OSPF and IS-IS as routing algorithms have nothing to do with the issues of ISO and IP suites. OSPF and IS-IS mostly differ in packet encoding procedures, control of large packets, database descriptions, area partitions, and router elections. In general, these two protocols have a lot of engineering trade-offs [14].

Even though there are several differences, we mention two of them which are related with the algorithmic search of the shortest path. Firstly, the neighbor initialization protocol is not the same for OSPF and IS-IS. In OSPF, when two routers become neighbors, it has to be defined a "master-slave" mechanism, in which the master transmits all parts of a package with explicit acknowledgment of what has been lost and what has been delivered as expected. On the other hand, in IS-IS, there is a higher level of tolerance when fragments of packages are lost or when they get delivered out of order.

Secondly, as it has already been mentioned, the "hello" messages facilitate the election of the neighbors for each router. In OSPF, the election procedure is more complicated than the one in IS-IS, which makes it more disruptive to have a change of which router has the highest priority when the "hello" messages are exchanged in OSPF. The fact that OSPF involves the decision making between routers, in order to define their roles as "master" and "slave" makes this process slightly slower. The master transmits a package one at a time and then the slave replies back and there is no possibility of pipelining. Moreover, in OSPF each part of the package must be delivered and this protocol requires explicit acknowledgment. On the other hand, the higher tolerance of IS-IS, in addition to a better mechanism when the database gets overloaded are two characteristics that makes IS-IS a better option when we want a protocol that overcomes networking problems by itself. For instance, IS-IS includes mechanisms with which a temporary database overload will be resolved by itself without human intervention, since an overloaded router can be treated as an end point in order to remain reachable. All in all, a package that has to be delivered from a source to an end node via the optimal path will be delivered both with IS-IS and OSPF. The difference is that with OSPF there is a higher probability to transmit all the required parts of the package with increased safety and a better routing optimization, whereas with IS-IS the transmission is guaranteed even if networking problems occur and some package parts are lost, as well as it is less memory intensive. [14].

But why is OSPF used in CERN's network? What are those characteristics that makes it a good routing protocol for this specific computer network? First of all, the Large Hadron Collider beauty (LHCb) experiment specializes in detecting slight differences in matter and antimatter by studying a specific particle called "beauty quark", or else "b quark". As a result, there is a high requirement of safety when one package is delivered from one router to another. OSPF's mechanism of identifying and delivering again and again packages that have not been transmitted properly to the destination enhances the safety that is required. Secondly, based on a specific research [16], some routing protocols' parameters have been evaluated, such as the cost of transmission, the network throughput (the amount of data that can be transferred from source to target within a specific time frame), as well as the delay. OSPF has the least cost of transmission and the maximum network throughput. Regarding the queuing delay, OSPF has small delay compared to the other two protocols in the research (RIP and IGRP). Thus, these three characteristics seem ideal in a huge network with high requirements of speed, time, and power.

The most important feature of OSPF though is that in the OSPF protocol there is a routing strategy for saving energy in computer networks [6]. Since in such a large experiment as the LHCb, the power consumption is approximately tremendous, the energy saving characteristics of OSPF enhance its advantages. In particular, this mechanism reduces the set of active network links by the utilization of a Shortest Path Tree between the neighbor devices. This is why this energy saving solution makes OSPF a "Green Protocol" [4] that should be used in huge computer networks, where the power consumption falls to about 80 megawatts.

Now that the usage of OSPF has been motivated, we dissect how it works. To begin with, each device has a copied graph of the complete network's topological map of the whole AS. Then, each device runs a path finding algorithm to decide the shortest-path till the target device, which is the final router to which the information package has to be sent. Moreover, every device is connected with all the others through weighted links that have been configured by the network administrator. Particularly, the path algorithm that is running now in OSPF in CERN's network is Dijkstra. Therefore, the main interesting point of OSPF, as regards to this research

project, is that it is a link-state protocol that uses Dijkstra's least-cost path algorithm. Given that, it must be investigated if there is another alternative to Dijkstra that could outperform what is already and standardly being used. One option is to replace Dijkstra with the A* search algorithm, which is an extension to Dijkstra's algorithm by using a heuristic. Due to that, the first question that has to be answered is if A* could outperform Dijkstra by replacing it in the OSPF protocol under the same constraints.

Before we give the description of OSPF in a planning format, we should refer to its important concepts based on the operating system of the devices. The operating system used in CERN's network is the Junos OS. In this Juniper network the OSPF protocol is enabled on all interfaces. The minimum device configuration for a device in an OSPF network is the following:

1. OSPF Areas: An area in OSPF is a group of routers, or else a sub-domain of routers, links, and devices that have the same area identification. The most important area is the "area 0" or "backbone area", which is connected with all the other areas in the network. Therefore, the configuration of this area is part of the basic configuration.

2. OSPF interfaces: Each of the devices establishes connections and communicates with the others through interfaces. The configuration of the interfaces within an OSPF area contributes to enable the OSPF protocol.

3. OSPF router identifiers for every device in the network.

Despite these minimum configurations, there are many more advanced ones that are usually needed, like authentication, links, routing instances, timers, routing policy, restarts, alternate routes, support for traffic engineering, fault detection, route control, and database protection.

1.1.4 What is planning in AI

The planning problem is about the way that intelligent machines make decisions in order to execute a task and achieve a goal. In the simplest planning problem in Artificial Intelligence we have a deterministic, real-world environment with one agent which is trying to complete a specific task. The agent has a unique known initial state and follows an acceptable sequence of actions, but it does not have complete information about the environment. If we want to dive into the specifics of the search problem, we ought to examine the heuristic evaluation function, which gives the cost of a specific sequence of actions until the goal has been achieved under certain constraints. These constraints are restrictions for the values of the variables that are being used. In addition, taking into consideration the time complexity of the search problem, we have to minimize the cost that is calculated from the heuristic evaluation function [8]. We emphasize that taking advantage of the heuristic functions and of the information from previous searches is what can be proved beneficial in planning, or else why planning can outperform traditional graph-search algorithms.

As it has already been mentioned above, the planning applies to a search problem, which is a repetitive process. This means that one agent solves the search problem by performing several iterations, executing the deterministic actions, and then replanning based on the results. A great assumption in all these steps is that the changes are usually small and have no significant impact in between these iterations [7]. All in all, in the simplest path-planning problem in Artificial Intelligence, the final goal can be achieved and the state of the world can be predicted accurately in the end because this unique initial state is always known and all actions are deterministic.

Taking into account the above information, a planning problem in AI is represented by logical sentences that describe the aforementioned main parts of a problem, with reference to [13, Chapter 11]:

1. **Initial state:** A logical sentence about the initial situation T_0. For the path-finding problem this could be
$$At(start, T_0), \neg goalFound, cost = 0$$

2. **Goal state:** A logical query, which defines the end of the repetitive process of planning. For the path-finding problem this could be
$$At(end, T), goalFound = True, cost = x, x \in (0, \infty)$$

3. **Operators:** A set of the actions, which have a specific representation. For example, a state representation in order to move to the next available node in a simple path-finding problem, like Dijkstra's, is represented as
$$MoveTo(from, to) \iff$$
$$In(from, S) \wedge In(to, V - S) \wedge IsEdge(from, to) \wedge min(cost + CostWeight(from, to))$$
where S is a set of the explored nodes in the graph and the chosen node from the unexplored part of the graph should have at least one edge from S which minimizes the path cost [15].

In order to put into context the operators, we should explain which three parts are included. In particular, there is the action description, the precondition, and the effect. For example, if we consider the latest action we have the following components, according to [13, Chapter 11]:

- The action description is the name of the corresponding action.
- The preconditions are logical sentences that must be true, so as to apply the operator.
- The effect shows what is actually being changed when the operator has been applied.

In other words, a simple definition of an operator is the following:

Operator(ACTION: MoveTo(from, to),
PRECOND: In(from, S)$\wedge$ In(to,V-S) $\wedge$ IsEdge(from,to) $\wedge$ min(cost+CostWeight(from,to)),
EFFECT: In(to, S), cost=min(cost))

1.1.5 The OSPF protocol in the context of planning

In this point, we have to describe the OSPF protocol in the context of planning. A simple planning representation of OSPF can be obtained by translating the properties of this protocol into a planning format. To start with, a path-finding algorithm is used in OSPF to search for the shortest path from a current device to a destination device. In particular, a search problem is transformed into a graph $G = (V, E)$, where the nodes $v \in V$ represent the devices of the network, while the edges $e \in E$ represent links between all these devices. In other words, the graph is a map topology which represents the computer network. What is more, the actions' costs and the problem constraints are encapsulated in the graph. Moreover, there is a single agent, which is our source node in the graph, or else the device that wants to send a package into another machine in the network. So, the solving strategy is based on this single agent and

the computer network is represented as a graph of nodes, while the search algorithm is the core of the OSPF protocol, and based on what has already been discussed, it could be the Dijkstra algorithm, or A*. In addition, an action is a movement to a new device (node) in the network (graph) which is working properly by sending and receiving packages.

With reference to the solving strategy, notice that by applying the search algorithm, either Dijkstra or A*, we build the final path from source to target node $s = (s_1, s_2, ..., s_i)$, where s_i represents the target device/or router in a specific OSPF area. Taking into consideration the cost of each link between two devices, we select the best route to the destination. The "Cost" is the value of the metric in the OSPF protocol. In particular, this is the reference bandwidth divided by interface bandwidth which varies among different interfaces. As a result, there is a need of a search algorithm that finds the best route regarding the "Cost" values of the links in the most efficient way. The main purpose is to minimize the "Cost" in such a problem, e.g. if the cost is the distance between two nodes or the time delay, we aim to achieve a package transmission as soon as possible. With respect to the actions, there are two planning actions. The first action is a movement to the next node based on the cost of all the adjacent nodes, whereas the second action is the calculation of the cost of the final path.

A different approach of the simple planning representation of OSPF can be obtained by defining and explaining the following planning terms:
Objects: The things in the world that interest us in this search problem are the devices of the network and the links between them.
Predicates: The properties of these objects are related with the bandwidth of the links, the speed that the packages are transmitted in the network, the routers that are working properly, as well as the network topology so as to understand the neighbors of each node and the adjacent links.
Initial state: All links have a specific "Cost" value, whereas specific devices that are out of order are labeled as "obstacles" in the path finding process. Also, for the search algorithm, we must know the cost of each link for every node.
Goal specification: We must find the optimal route or the best path from a source to a target node.
Actions: The movement to the next node based on which one optimize the cost for the final path.

Finally, we can explicitly describe the search problem of OSPF, which runs the Dijkstra's algorithm, as a planning problem as follows:

1. **Initial state** in the deterministic environment of our network, which is represented by a graph $G = (V, E)$, where the nodes V represent the devices of the network, while the edges E represent the links between all these devices. Let's assume that $s = startingPoint$, S is a set of the explored nodes, T_0 is the state at time=0 and $d(s)$ represents the path cost:

$$At(s, T_0), d(s) = 0, S = \{s\}$$

2. **Goal state:** The goal is to transmit a package of information to the target node, which is another active device in the network. We assume that $z = endPoint$ and T is the final state

$$At(z, T), d(z) = \min d(z), \forall z \in V - S$$

3. The **actions** are the following:

(a) We choose a node that has not been "explored" yet

Operator(ACTION: ChooseNode(v),
PRECOND: At(v∈ V-S),
EFFECT: At(v∈ S))

(b) Choose a single edge towards the explored part S:

Operator(ACTION: ChooseEdge((u,v)),
PRECOND: At(v∈ V-S) ∧ At(u∈ S),
EFFECT: At(v∈ S))

(c) Check that this edge satisfies the main constraint of the problem by minimizing the path cost $d'(v)$. Assume v is the current node:

Operator(ACTION: CheckCost((u,v)),
PRCOND:d'(v)=$\min_{e=(u,v):u\in S} d(u)+l_e, \forall(u \in S)\wedge\forall(v \in V-S)\wedge\exists(u,v)$,
EFFECT: d(v)=d'(v))

(d) Update the set of explored nodes:

Operator(ACTION: Add(v,S),
PRECOND: ,
EFFECT: At(v∈ S), ¬ At(v∈ V-S)

(e) Check termination:

Operator(ACTION: Exit(V,S),
PRECOND: S=V,
EFFECT: Exit(), ¬ Search()

References

[1] Abbie Barbir, Sandra Murphy, and Yibin Yang. Generic threats to routing protocols. *IETF Draft: draft-ietf-rpsec-routing-threats-07*, 2004.

[2] Adi Botea, Martin Müller, and Jonathan Schaeffer. Extending pddl for hierarchical planning and topological abstraction. In *In Proceedings of the ICAPS-03 Workshop on PDDL*. Citeseer, 2003.

[3] Antonio Capone, Carmelo Cascone, Luca G Gianoli, and Brunilde Sanso. *OSPF optimization via dynamic network management for green IP networks*. IEEE, 2013.

[4] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Marazza, and Enrico Vittorini. An energy saving routing algorithm for a green ospf protocol. In *2010 INFOCOM IEEE Conference on Computer Communications Workshops*, pages 1–5. IEEE, 2010.

[5] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, and Marco Polverini. An ospf enhancement for energy saving in ip networks. In *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 325–330. IEEE, 2011.

[6] Antonio Cianfrani, Vincenzo Eramo, Marco Listanti, Marco Polverini, and Athanasios V Vasilakos. An ospf-integrated routing strategy for qos-aware energy saving in ip backbone networks. *IEEE Transactions on Network and Service Management*, 9(3):254–267, 2012.

[7] Sven Koenig, Maxim Likhachev, and David Furcy. Lifelong planning a*. *Artificial Intelligence*, 155(1-2):93–146, 2004.

[8] Richard E Korf. Planning as search: A quantitative approach. *Artificial intelligence*, 33(1):65–88, 1987.

[9] James F Kurose and Keith W Ross. Computer networking, 1991.

[10] Dale Liu. *Cisco CCNA/CCENT Exam 640-802, 640-822, 640-816 Preparation Kit*. Syngress, 2009.

[11] John T Moy. *OSPF: anatomy of an Internet routing protocol*. Addison-Wesley Professional, 1998.

[12] Alexander Nareyek, Eugene C Freuder, Robert Fourer, Enrico Giunchiglia, Robert P Goldman, Henry Kautz, Jussi Rintanen, and Austin Tate. Constraints and ai planning. *IEEE Intelligent Systems*, 20(2):62–72, 2005.

[13] P Russel Norvig and S Artificial Intelligence. *A modern approach*. Prentice Hall, 2002.

[14] Radia Perlman. A comparison between two routing protocols: Ospf and is-is. *Ieee Network*, 5(5):18–24, 1991.

[15] Eva Tardos and Jon Kleinberg. Algorithm design, 2005.

[16] V Vetriselvan, Pravin R Patil, and M Mahendran. Survey on the rip, ospf, eigrp routing protocols. *International Journal of computer Science and information Technologies*, 5(2):1058–1065, 2014.