SEARCH-TREE BASED SDN CANDIDATE SELECTION IN HYBRID IP/SDN NETWORK

NING LI

ASSISTANT PROFESSOR

HARBIN INSTITUTE OF TECHNOLOGY (HIT)

1. Introduction

1.1. Motivation

1 Link failure recovery is important to the Internet since it guarantees to provide reliable and secure service for the user.

2 The previous solutions have disadvantages on addressing link failure recovery in IP network; such as cannot protect all link failures, the existence of the repair path.

3 The SDN is effective on improving the performance of link failure recovery; however, they have disadvantages: (1) for the greed-based algorithms, only one feasible solution can be found each time; (2) the greedy-based approach cannot guarantee to find the optimal solution always; (3) for a certain link, more than one candidate SDN switches can protect the failure of this link, how to choose the most appropriate one as the final designated SDN switch has not been investigated. .

1. Introduction

1.2. Contributions

1. In this paper, we propose the search-tree based SCS algorithm. Based on this algorithm, all the feasible solutions can be found. Moreover, according to the branch and bound, the complexity of the search tree is reduced.

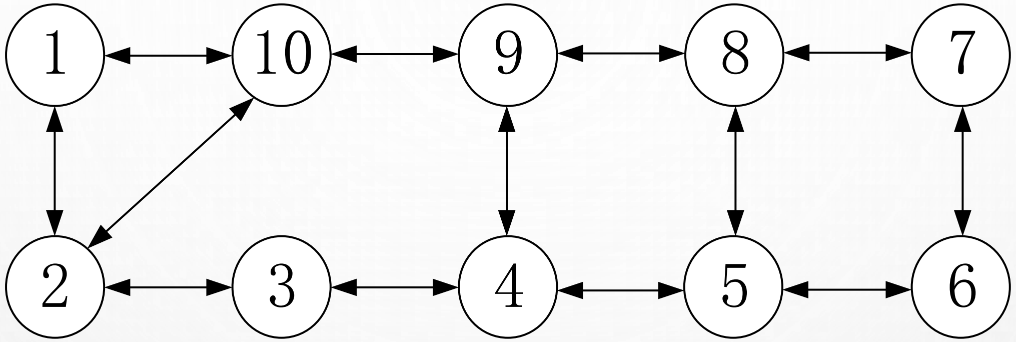
2 Since more than one feasible solution can be found, we proposed the most appropriate feasible solution selection algorithm. In this algorithm, the reliability degree of each feasible solution is defined and the most appropriate feasible solution is chosen based on it.

3 Considering that for each link, there is more than one SDN switch can protect it, we propose the most appropriate designated SDN switch selection algorithm to select SDN switch for each link. In this algorithm, the average repair path length and the average link utilization of each SDN switch are considered; .

4 We compare the search-tree based algorithm with the greedy-based algorithm, the simulation results show that the search-tree based algorithm can improve the performance greatly.; .

2. Network model and Problem statement

2.1. Network model



1 $G = (V, E)$, where V represents the number of nodes and E represents the number of bidirectional links; (i, j) is the bidirectional link.

2 When link (i, j) fails, both node *i* and node *j* can detect this event. For recovering the failure traffic, node *i* and node *j* will reroute the data flow in both link $< i, j >$ and link $< j, i >$.

3 In hybrid IP/SDN network, the conditions that a designated SDN switch can protect the failure of a directed link are defined in [10] and [11]: (1) the shortest path from router *i* to SDN switch *k* does not include link $< i, j >$ (where $< i, j >$ is the failed link); (2) for each affected destination, there exists at least one next-hop *h* of switch *k*, and the shortest path from *h* to the affected destination does not include $< i, j >$.

2. Network model and Problem statement

2.2. Problem Statement

Link Failure	SDN candidate									
	1	2	3	4	5	6	7	8	9	10
1, <1,2>	1	0	0	0	0	0	1	1	1	1
2, <1,10>	1	1	1	1	1	1	0	0	0	0
3, <2,1>	0	1	0	1	1	1	1	1	1	1
4, <2,10>	1	1	1	1	1	1	0	0	0	0
5, <2,3>	0	0	0	0	0	0	1	1	1	1
6, <3,2>	0	0	0	1	1	1	1	1	1	0
7, <3,4>	1	1	0	0	0	0	0	0	0	1
8, <4,3>	0	0	0	0	0	0	0	0	1	1
9, <4,5>	0	0	0	0	0	0	0	0	1	0
10, <4,9>	1	1	1	0	1	1	1	1	0	0
11, <5,4>	0	0	0	0	0	0	0	1	0	0
12, <5,6>	0	0	0	0	0	0	0	1	0	0
13, <5,8>	1	1	1	1	0	1	1	0	1	1
14, <6,5>	0	0	0	0	0	0	1	0	0	0
15, <6,7>	1	1	1	1	1	0	0	1	1	1
16, <7,6>	1	1	0	0	0	0	1	1	1	1
17, <7,8>	0	0	1	1	1	1	1	0	0	0
18, <8,5>	1	1	0	0	0	0	1	1	1	1
19, <8,7>	0	1	1	1	1	1	0	1	1	0
20, <8,9>	0	0	1	1	1	1	1	1	0	0
21, <9,4>	1	1	0	0	0	0	1	1	1	1
22, <9,8>	0	1	1	1	1	1	0	0	1	0
23, <9,10>	0	0	1	1	0	0	0	0	0	0
24, <10,1>	0	1	1	1	1	1	1	1	1	1
25, <10,2>	1	0	0	1	1	1	1	1	1	1
26, <10,9>	0	1	1	0	0	0	0	0	0	0

2. Network model and Problem statement

2.2. Problem Statement

1 as the example shown in Section II.B, except for $(8,3,9,1,7)$ and $(9,3,8,2,7)$, there are still many other feasible solutions, such as $(7,8,9,3,10)$ and $(7,8,9,2,4)$. Even the $(8,3,9,1,7)$, $(9,3,8,2,7)$ and $(7,8,9,3,10)$ can be gotten by executing the greedy-based approach repeatedly, the $(7,8,9,2,4)$ cannot be found based on both the approaches proposed in [10] and [11].

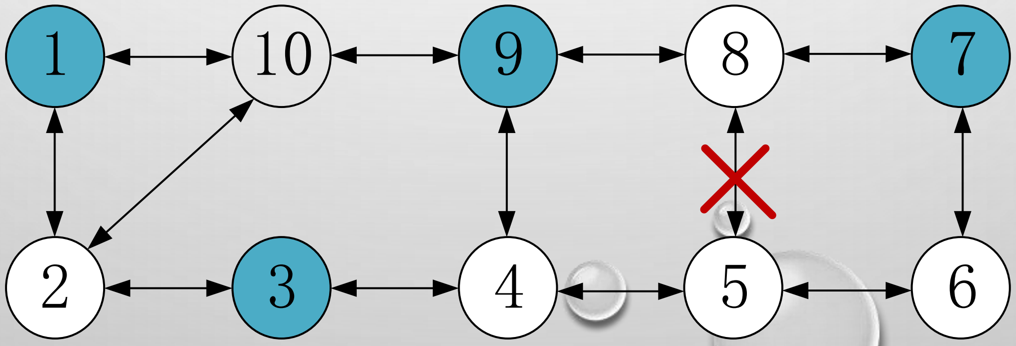
2 the greedy-based approach cannot guarantee to get the optimal solution always. For instance, as the example shown in Table 2, based on the greedy-based approach, the feasible solution should be $(3,2,1)$. However, intuitively, the $(2,1)$ is better than $(3,2,1)$ because: on one hand, it can meet the constraints shown in Section II.A, on the other hand, the number of SDN switches is smaller than $(3,2,1)$.

Link failure		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SDN candidate	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	3	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	4	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0
	5	0	1	0	0	0	1	0	0	0	1	0	0	1	0	1
	6	1	0	0	0	0	0	0	1	0	0	0	1	1	1	1

2. Network model and Problem statement

2.2. Problem Statement

3 for each link, there may be more than one SDN switches can protect the failure of this link. For instance, as shown in Fig. 2, for the solution (8,3,9,1,7) calculated by the greedy-based approach, the link $< 5,8 >$ can be protected by four SDN switches, i.e., SDN_1, SDN_7, SDN_3, and SDN_9. In previous works, when link $< 5,8 >$ fails, which SDN switch is used to reroute the data flows in this link has not been investigated. So, the properties of the SDN switches, such as the repair path length and the load balancing, are not considered. Even the repair path length from the failed link to the specific destination through SDN switch *k* is calculated in [11], they use the average repair path length of each SDN switch to choose candidate column when there is a tie.



2. Network model and Problem statement

2.2. Problem Statement

(1) how to find all the feasible solutions with low computation complexity

(2) how to choose the most appropriate solution as the final SDN switches deployment locations;

(3) how to guarantee the proposed approach can always find the optimal solution;

(4) how to choose the most appropriate designated SDN switch for each link failure. These issues are not investigated by the previous works.

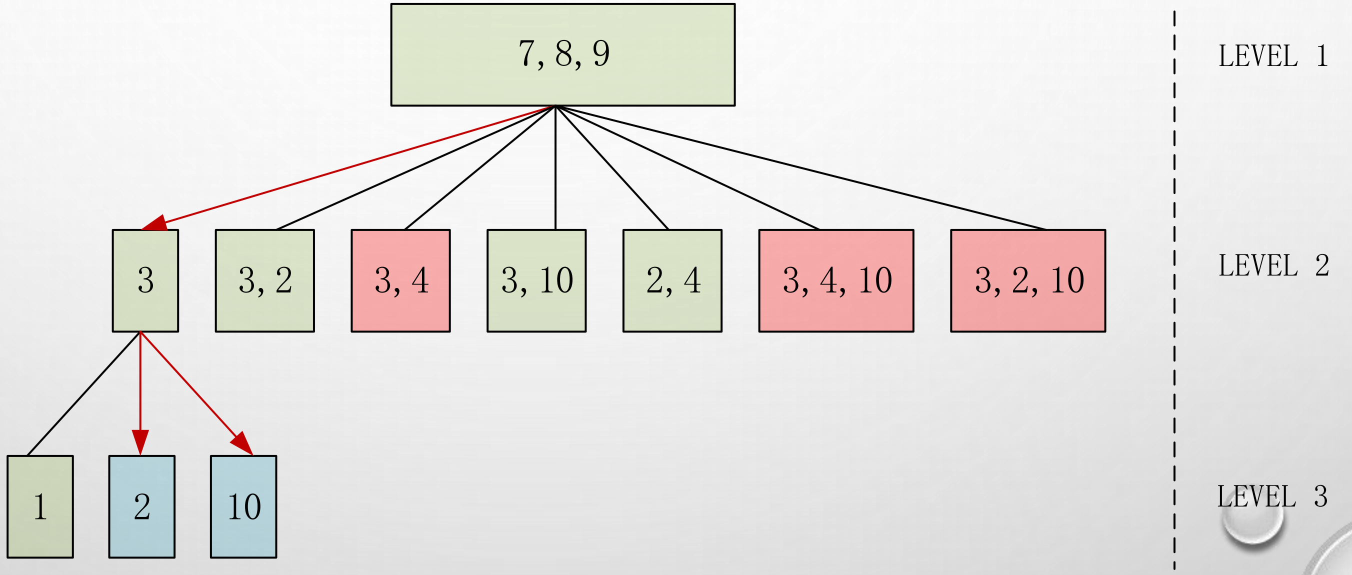
3. Proposed Search-Tree Based Approach

3.1. Search Tree based SCS algorithm

Link failure	Cover set	Reliability degree
1, <1,2>	1, 7, 8, 9, 10	5
2, <1,10>	1, 2, 3, 4, 5, 6	6
3, <2,1>	2, 4, 5, 6, 7, 8, 9, 10	8
4, <2,10>	1, 2, 3, 4, 5, 6	6
5, <2,3>	7, 8, 9, 10	4
6, <3,2>	4, 5, 6, 7, 8, 9	6
7, <3,4>	1, 2, 10	3
8, <4,3>	9, 10	2
9, <4,5>	9	1
10, <4,9>	1, 2, 3, 5, 6, 7, 8	7
11, <5,4>	8	1
12, <5,6>	8	1
13, <5,8>	1, 2, 3, 4, 6, 7, 9, 10	8
14, <6,5>	7	1
15, <6,7>	1, 2, 3, 4, 5, 8, 9, 10	8
16, <7,6>	1, 2, 7, 8, 9, 10	6
17, <7,8>	3, 4, 5, 6, 7	5
18, <8,5>	1, 2, 7, 8, 9, 10	6
19, <8,7>	2, 3, 4, 5, 6, 8, 9	7
20, <8,9>	3, 4, 5, 6, 7, 8	6
21, <9,4>	1, 2, 7, 8, 9, 10	6
22, <9,8>	2, 3, 4, 5, 6, 9	6
23, <9,10>	3, 4	2
24, <10,1>	2, 3, 4, 5, 6, 7, 8, 9, 10	9
25, <10,2>	1, 4, 5, 6, 7, 8, 9, 10	8
26, <10,9>	2, 3	2

3. Proposed Search-Tree Based Approach

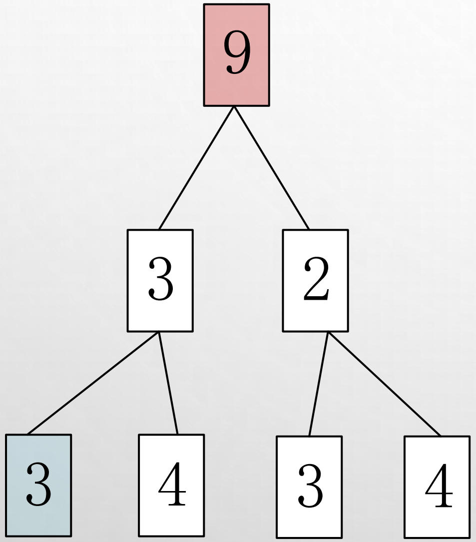
3.1. Search Tree based SCS algorithm

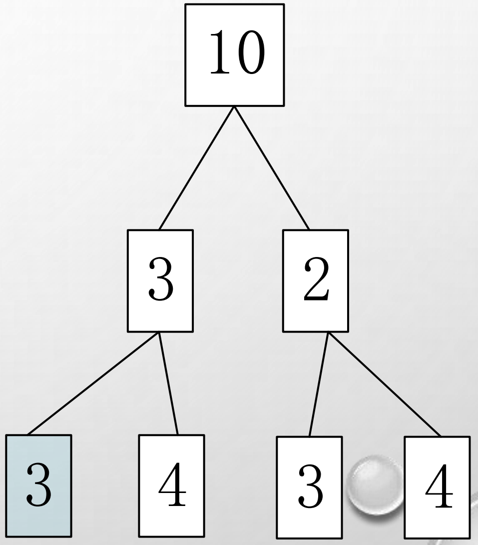


3. Proposed Search-Tree Based Approach

3.1. Search Tree based SCS algorithm

The SDN witches in the second level are: (2,3), (3,4), (9,10)





3.2. Most Appropriate Solution Selection Algorithm

Definition 1. The reliability degree of feasible solution i is defined as the number of SDN switches which can protect more than one link failures in i, denoted as rd_i.

$$v_1 = (4,2,3,2,3,3,1,1,1,4,1,1,4,1,4,3,2,3,3,3,4,2,1,4,4,1)$$
$$v_2 = (3,2,4,2,3,3,1,1,1,4,1,1,4,1,4,3,2,3,4,3,4,3,1,5,3,2)$$
$$v_3 = (4,1,4,1,4,3,1,2,1,3,1,1,4,1,4,4,2,3,3,3,4,2,1,5,4,1)$$
$$v_4 = (3,2,5,2,3,4,1,1,1,3,1,1,4,1,4,4,2,3,4,3,4,3,1,5,4,1)$$

where v_1 is the reliability degree vector of (7,8,9,3,1), v_2 is the reliability degree of (7,8,9,3,2), v_3 is the reliability degree of (7,8,9,3,10), and v_4 is the reliability degree of (7,8,9,2,4),

3. Proposed Search-Tree Based Approach

3.3. Most Appropriate SDN Switch Selection Algorithm

1. Average Repair path length

$$d_e^i = \sum_{j=1}^{m} d_{e,j}^i / m$$

2. Load balancing degree

$$r_e^i = \sum_{j=1}^{m} r_{e,j}^i / m$$

3. Utility of each feasible SDN switch

$$p_i = \omega_d^* d_e^{i*} + \omega_r^* r_e^{i*}$$

4. Performance evaluation

4.1. The number of SDN switches under different network topologies

	NSFNet	Abilene	Internet 2	40-node ER (0.1)
Greedy-based	3	5	5	4.4
Search-tree based	3	5	5	4.1

4.2. The reliability degree under different network topologies

	NSFNet	Abilene	Internet 2	40-node ER (0.1)
Greedy-based	13	6	5	11
Search-tree based	15	9	9	14

4. Performance evaluation

4.3. The number of SDN switches under different network topologies

	NSFNet		Abilene		Internet2		40-node ER (0.1)	
	d_e^i	r_e^i	d_e^i	r_e^i	d_e^i	r_e^i	d_e^i	r_e^i
Greedy-based	4.42	0.897	4.22	0.92	4.24	0.915	4.5	0.886
Search-tree based	4.18	0.82	3.89	0.837	3.91	0.741	4.23	0.802

Thank you for your attention!