1. Good morning, teachers. My name is Lihao from nanjing university. It’s my pleasure here to introduce my topic. My topic today is “A Greedy Heuristic Deployment strategy for VNFs”.
2. Network functions which is deployed on the physical network, which is called middle box, plays a crucial role in current network. There are some middlebox instance, such as Firewall, DDoS protection, IDS, ad insertion, BRAS. But Hardware middleboxes have many drawbacks. for example: Expensive equipment/power costs, Difficult to add new features, Difficult to manage, Cannot be scaled on demand. However, by separating network function from the underlying hardware, the deployment will be more flexible. So we can deploy network function in virtual network. And how to deploy dynamic, flexible and robust service functions is an attractive and useful problem.
3. So what we want to taik today is: Resource Scheduling for Network Function Virtualization. There gives the instance: given which denotes a physical network with fixed node resources and edge resources. And a service function chain set.. /\*In this figure, a,b,c,d,e,f,g denotes the nodes, and service function will be deployed in these nodes. \*/ denotes the ith services function chain want to be deployed in the physical network. And denotes a sequential service function order. And a service function will correspond to its neighbor, which will cost bandwidth resources. Our aim is to Find an optimal strategy to deploy the SFC set in the physical network.
4. Next is the related work. Resources Allocation of SFC. Virtual Machine Deployment. And Virtual Network Embedding.
5. Resources Allocation of SFC. Deploy virtual nodes to the physical nodes if the node computing resources and edge bandwidth is enough. Input: a physical network and a SFC set. Output: a network function deployment. This problem is similar to the problem we talk today, and the difference between them are: first, A physical node can be deployed with many functions in this problem. And in schedule problem, a node can be only deployed with one or two functions. Second, No share function node. And in schedule problem, we need to take function share in to consideration. Also, the problem is Fixed requirements. but we focus on a floating requirements.
6. Virtual Machine Deployment. Users requirement are deployed in the virtual machines, and virtual machine are deployed in the physical network. And it is similar to our problem. In our problem, the service function are deployed in virtual nodes. The problem of virtual machine deployment is to find a suitable place for virtual machine. So this problem is the next step of our problem.
7. Virtual Network Embedding. Embedded a virtual network to a physical network. In this figure, we need to embed the upper virtual network to the physical network. It also take allocation of node resources and edge resources into account. (here is the simplify ) and the differences is: it is an undirected cyclic graph. And we talk today is a DAG.
8. For multi-service chains, to find a optimal deployment plan is a NP-hard problem. So we do some assumptions: first, the service chain is queued to be deployed. In fact, in real environment, the service chain will seldom flourish to reach the queue. And if this situation occurs, we can schedule them one by one. Under common circumstances, a virtual machine can be deployed with a VNF function, and different service chains can share the same VNF function. So we can model our aim as this. We want to *Maximize the* number of the SFCs which can be deployed in the network. And we want to *Minimize the cost we need,* including the node computing cost and the bandwidth cost.
9. And the deployment mechanism is: Deployment of functions, optimal nodes set will be selected to load all the required VNF because virtual functions of different nodes in the physical network are different. And Selection of path: after selection of nodes, a proper path between nodes is also needed to make one node corresponding with another and achieve a splendid working efficiency. So, For an incoming VNF chain requests, we need to map all the VNFs of the service chain and construct a physical path to link all the nodes.
10. As more and more service function chain get into the network, and more and more node has been deployed with service function, and for a new coming service function chain, we wanna to find a node chain to deploy, we have 2 choices: 1. select a empty node, deploy the network function in the node. The other. reuse some nodes if the rest computing resources is enough. if the first choice is used, the whole physical nodes number is high which will increase the cost of nodes of the whole network. If the second choice is used, the cost of the edge will increase. And How to balance reused nodes and bandwidth costs is a considerable problem!
11. Let’s see an example. For a SFC , which has A,B and C functions. Node d has been deployed with A, node e has been deployed with B, node f has been deployed with C. and node a,b,c are empty nodes. If we use the empty node to deploy the service function chain, the bandwidth is bandwidth(AB)+bandwidth(BC). And if we use d,e,f to deploy the service function chain, the bandwidth is 2\*bandwidth(AB)+2\*bandwidth(BC). But there will be 3 empty node still to other functions to deploy.
12. The next point we consider is Dynamic requirements. The computing resources which is required by a virtual network function is not definite. And we can split the requirements into basic requirements and floating requiremens, and floating requirement occurs in a possibility of p. and we can also use statistic distribution to model this problem, and there, for simplicity，we use this model.

When we split the requirement into two parts, the next problem is: How to judge the share. For two functions, if they can share the node? We have the next comparison: RestCp() and there, we compare the rest computing resourses of the node and the whole requirements of the function and the crash possibility. If the crash possibility is lower than some fixed threshold, we approve the share. How to determine the threshold is also a attractive question, but it is not we wanna to talk today.

1. See a example, the fixing requirement of node **a** in R1 is 10 and the floating requirement of node a is 5 and the possibility is 0.3. the fixing requirement of node **a** in R2 is 10 and the floating requirement of node a is 5 and the possibility is 0.3. If they want to deploy in a node with 25 computing resources.
2. The next problem which is the core problem is: How to reuse physical nodes? Then we model the problem, X represent the number of node which will be reused for chain . X is dynamic. *l(x) the length of the edge when reusing the x nodes. l(x) depends on x.*

For a new coming service chain, it is hard to find its optimal reused factor, and we can transmit a point to solve the problem, assume there are infinite same service function chains, and find a x and l(x) to make the number which can be deployed in the network maximum. The transform based on the fact that compared with service function chain, the node resources and edge resources of the network is numerous, and if we reuse some nodes of SFC in this step, for a next same function chain, there will also be reused nodes. Then we get the MP rules. And the comparison of the computing resources and bandwidth resources is a vector comparasion, for all edge and all nodes, there will be a comparison.

And the length can be defined by x.

1. After find the reused factor, then we can split a chain to sub-chains by the reuse node. in this example, we can divide the chain into 3 parts: a-b-c,c-d-e,e-f. And we can deploy the sub-chain and combined all the sub-chains. The choice of nodes depends on the network characteristics.
2. Then we get a heuristic deployment plan: first, Find the reused function x for an incoming VNF chain. Then, Divide a service chain into (x+1) sub-chains. Last, Greedily find an optimal deployment for each sub-chain, and finally deploy the entire VNF service chain.
3. And then we get the greedy algorithm, input: D：service chain which represent requirements set, and physical network graph. And output: if the requirement can be satisfied, then return the deployment of D in G.
4. Then we use Python networkX to design a simulation deployment model. And get these results. In this figure, two strategy cannot achieve the optimal result. But use greedy algorithm is better than the predict heuristic algorithm. The second figure shows when the node connection rises, the result of the two algorithm lifts obviously. And the difference between the two algorithm is reduced. Both of them get a good result. And the third figure compared the algorithm with dynamic requirements and without dynamic requirements, and the possibility of the successful deployment lifts obviously, which proves the correct of the dynamic requirement.
5. Conclusion: today, we talked with Resource Scheduling for Network Function Virtualization Dynamic requirements Reused nodes Dynamic programming. And the aim of our topic is : ensure the stability of the network and also increase the efficiency
6. Thank you !