The Implementation of Virtualization in Data Plane of ForCES

Zou Xi
College of Information & Electronic Engineering
Zhejiang Gongshang University
No.18, Xuezheng Str., Xiasha University Town, Hangzhou, P.R. China, 310018
Email: 387974178@qq.com

Gao Ming, Weiming Wang
College of Information & Electronic Engineering
Zhejiang Gongshang University
No.18, Xuezheng Str., Xiasha University Town, Hangzhou, P.R. China, 310018
Email: 497862918@qq.com

***Abstract*—In recent years, with the rapid development of virtualization technology, the emergence of network virtualization solves the problem of ossification. At the same time, it provides endless possibilities for the innovations of the network architecture. As a result, it has attracted the attention of the next generation Internet architecture. Therefore this paper based on the architecture of ForCES to explore the implementation of the data plane virtualization and provide the framework for virtualization platform via the method. Meanwhile, in the premise of meeting the requirements which the virtual network packet has enough processing capacity. For the sake of solving the allocation of FE resource in virtual network (Forwarding Element, FE). To make the number of FE and the utilization of FE more reasonable. This paper puts forward a kind of FE resource allocation algorithm based on twice iteration subtraction. We use Click Modular Router as the data plane processing engine, then we give the virtual method to realize data plane and explain that supporting multiple virtual network and the virtualization of FE can provide more flexibility for processing data packets.**

Keywords—ForCES; Network virtualization;

I. INTRODUCTION

Because of the unprecedented challenges in innovation, expansibility and management the traditional Internet has faced now. The problem about ossification completely unmasked [1].For example, the existing framework cannot solve the conflict between the network performance and network expansion; it's unable to adapt to the need of network of emerging technologies and extended architecture research; it also can't satisfy the development of diversified business and sustainable social demands. How to carry out the technical innovation on the Internet in the case of the existing network infrastructure architecture, what's more, it must do not affect the daily use of the Internet. And now it has become a research hotspot.

With the new network architecture represented by Openflow[2] has become a hot spot, more and more researchers pay attention to network virtualization based on the separation of control plane and data plane. Through the long-term research, we found that the flexible architecture design in ForCES (Forwarding and Control Element Separation) emphasizing the Loose coupling relationship between the Control Element (Control Element, CE) and the Forwarding Element (Forwarding Element, FE). This happen to hold the same view with Node virtualization research in Network virtualization. It is based on this point, we study how to implement the virtual node through ForCES extension and then realize the virtualization of the whole network. More coincidence is almost at the same time, IETF ForCES working group has also launched the discussion about "The effect of FEM and CEM in network virtualization".

IETF ForCES working group was established in 2001. The group has completed the ForCES demand analysis [4], the definition of the basic framework of ForCES [5], the definition of the model of FE [6], LFB library [7] and MIB library [8]. Now the group is devoted to promoting the research on the key technology of SDN based on ForCES. The core idea of ForCES technology is that the control plane and the data plane are separated from the traditional network equipment. To ensure that the network operator can flexibly recombine all functions of module and meet the different needs of the business, achieve the open and reconfigurable goal, the data plane should made the resource more modular and more standardized.

A network equipment which follows the rule of ForCES architecture is defined as the ForCES network element (Network Element, NE). A ForCES network includes a control element (Control Element, CE) and a plurality of forwarding element (Forwarding Element, FE), where CE also can have more than one for the sake of redundancy. ForCES protocol [9] provides information interaction between CE and FE. It is a core protocol supporting ForCES forwarding element and control element separation. RFC 5810 has defined the protocol specification. The internal structure of FE was defined by the model of ForCES FE protocol (RFC 5812). FE is mainly responsible for the rapid processing of data packet, such as packet encapsulation, classification, look-up table and scheduling. ForCES uses the modular idea to abstract the hardware and software resources which completed these actions in FE into a series of logical function blocks (Logical Function Block, LFB). Each LFB is composed by five part, input, output, properties, ability and events. Where input and output are the channel between LFB to make data pass in and out, and they have been determining the connection between LFB, the sum of connections have formed a set of FE-Intra topology. In contrast with the attention on the dynamic configuration of FE-Inter topological, FE model also focuses on dynamic FE-Intra topology. By controlling the ForCES protocol messages, and combining the different functions of LFB, so that it can directly affect the data processing steps of packets, and it shows that FE can dynamically change the method of packet processing.

II. FORCES-IN-FORCES ONE KIND OF NETWORK VIRTUALIZATION METHODS

The architecture of ForCES is designed to break the closed characteristics of the network equipment machine, so that network device has the characteristics of high modularization. Network node in the ForCES consists of a CE and a plurality of FE, and the combination is flexible. According to the characteristics of ForCES architecture, this paper tries to introduce the network virtualization technology. A number of CE and FE are virtualized in physical machines CE and FE, the virtual devices are called virtual control element (Virtual CE, vCE) and virtual forwarding element (Virtual FE, vFE). These vCE and vFE can realize the same functions as physical CE and FE. vCE can combine with one or multiple vFE to comprises a virtual network element(Virtual NE, vNE) in logic. Therefore, we can construct a plurality of vNE in a physical ForCES NE, as in NE, there is no special requirements for the physical location of the CE and FE, and they can be freely combined. Thus, NE itself is a logical concept. These vNE has been isolated from each other, independent from each other. This is because there is no interdependence between vCE and vFE, they were occupying the outer physical machine resources in exclusive mode. Figure 1 is the schematic diagram of the virtualization solutions.

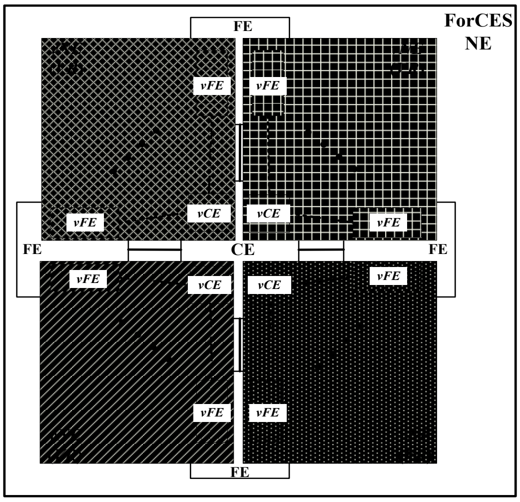


Fig. 1. The schematic diagram of ForCES-in-ForCES network virtualization method

When an NE on the need to run multiple virtual networks (Virtual Network, VN) at the same time, this NE can assign a dedicated VR (Virtual Router, VR) for each virtual network, these isolated VR guaranteed the isolation between the virtual network. In essence, the network virtualization method based on ForCES this paper has assigned is a kind of ForCES-in-ForCES architecture design.

The outer layer of the ForCES is the extension of traditional ForCES, it builds NE on the one hand that exhibits a physical network node as Forwarding and Control Element Separation. On the other hand, it also needs to strengthen the management of vFE and vCE. In order to control the construction process of vNE. The inner layer of ForCES, in the traditional sense, is ForCES. What is different about this is that the operating environment was changed into a virtual machine. This architecture of ForCES-in-ForCES has solved the problem of node virtualization, meanwhile, it also has solved the problem of network-level virtualization, so it is a novel and practical method of network virtualization.

Based on the method of ForCES-in-ForCES, we have designed and implemented a ForCES network virtualization platform (for narrative convenience, the following sections referred to as vForTER). vForTER is a cluster of physical servers, one of which is control plane device, named CE, data plane is typically dispersed on multiple servers, which perform the function of data forwarding, called FE. vForTER builds a virtual router VR for virtual network, and ensures that each virtual network corresponds to a VR. The number of VFE in VR depends on the desired forwarding capability of virtual network, and the available resources of vForTER.

III. THE MOTIVATION OF DATA PLANE FE VIRTUALIZATION

FE is the central component of data plane of vForTER. Here we first give the structure of FE, then research on the problem about FE support multiple virtual network. In the design of data plane virtualization, we have designed a VirtualBox virtualization platform, FE virtual Process Management (FEM), ForCES middleware configurator and Resource Pool. In addition, we also designed a Listener module to make CEM as a global manager, and control the creation of virtual forwarding element (Virtual FE, vFE).This paper will introduce the technical specifics of data plane virtualization in detail.

In vForTER, although VR monopolizes the use of FE to get better performance of packet forwarding. However, it can provide greater flexibility for processing data packets by FE virtualization or via supporting multiple virtual network.

In order to reduce the unnecessary overhead of system, the data plane of vFE typically operate in kernel state. Thus, the inadvertent error of packet processing may lead to the paralysis of the entire FE machine, as for this problem, it is especially true when VR monopolizes the use of FE. The only solution is to restart the machine, but it is a huge administrative overhead in large-scale network scenarios.

IV. THE REALIZATION OF VIRTUAL FE

A. The internal structure of the FE virtualization platform

It is automatic deployment that FE virtualization platform creates vFE (Virtual FE, vFE). The command of creating vFE can be assigned by CE Virtualization Process Management (CEM) module, or FE Virtualization Process Management (CEM) module. For the bottom of virtualization platform, we use Oracle VirtualBox as virtualization software.

VirtualBox is open source software used as virtual machine, users can install or run Solaris, Windows, DOS, Linux, OS / 2 Warp, BSD and other systems as a client operating system on VirtualBox. In contrast with Vmware Workstation, VirtualBox is more compact, less system resource occupation, and it can run more stably and smoothly. In addition, VirtualBox supports the management of virtual machines through the command line, it comes with VBoxManage command can create, delete, start, stop the virtual machine, and it also can modify the configuration options of virtual machine. Figure 2 shows the internal structure of the FE Virtualization Platform.

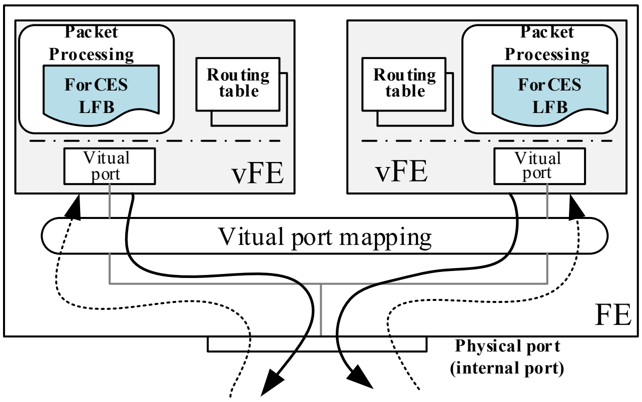


Fig. 2. The internal structure of FE

In this virtualization platform, the resources of the outer FE can be partitioned into a set of logically detached vFE. The data plan of every VR (Virtual Router, VR) was composed of one or more vFE. After the data packet getting into the Virtual ForCES Router, It will be further processed by vFE. The functions of vFE Packet Processing was provided by the ForCES Logic Function Block ((Logical Function Block, LFB). LFB provides a unified method for FE Packet Processing, meanwhile CE can control the underlying hardware and software resources by operating these LFB. Packet processing module includes packet classification, packaging and address look-up table and other operations. The routing table of vFE was maintained by the virtual control element (Virtual CE, vCE). In order to make the corresponding port of vFE receive data from the specified physical port, the port on the need to set as a bridging mode, and bound to the corresponding physical port.

B. Packet Forwarding Engine

In the traditional ForCES architecture, the packet forwarding of FE can be completed by the Linux kernel, but this way may affect the efficiency and reliability of system. So this paper decides to use the Click Modular Router as packet forwarding engine.

Click is a software routing system designed by Dr. Eddie Kohler from MIT University. It is a new software architecture for building flexible and configurable routers. Click can process packets in a modular way, and it's able to run on ordinary PC, server and part of the MIPS architecture, as the forwarding engine of data plane in a router. A complete Click Modular Router should have three parts, including component elements, packet structure and the way of connection.

Click has two modes during operation, respectively is the user mode and kernel mode. When click runs in the user mode, the Packet processing module running in user space. After data packets being processed, then they need to be copied from user space to kernel space to complete forwarding. While running in the kernel mode of Click, data packets don't need to be copied, in contrast, data packets will be directly forwarded after processing. When comparing these two modes, you can find the former with big time cost, low efficiency, but stable operation, and the later with high efficiency, but unstable, prone to errors. Therefore, Click runs in which mode depending on the specific needs of the situation.

In this platform, Click as the underlying data forwarding engine of vFE, If Click is regarded as the bottom of the forwarding hardware, LFB is the abstraction layer between the application and hardware. The control element will increase or decrease LFB or modify the value of LFB, based on packet processing demands of virtual network, so as to achieve the purpose of modifying the underlying respective components of Click. We designed a configuration file and wrote scripts for Click, and modified the unicast of vFE which longest prefix matching LFB and next hop LFB, so as to make it possible for LFB to modify the Click configuration files by executing scripts. However, what are really modified is static routing components, the final purpose is to manually increase or decrease the routing. Figure 3 shows the connection of Click router elements used in this platform.

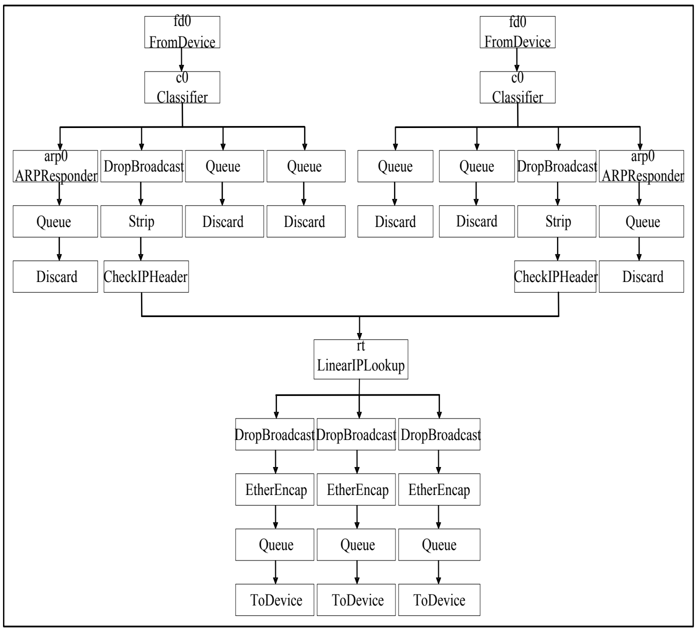


Fig. 3. The connection of Click router elements

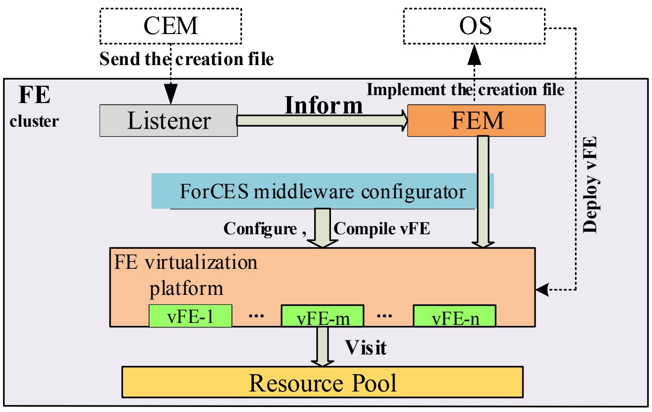
Fig. 4. The relationship between each module of data plane

The core element of this configuration file is “LinearIPLookup”, whose function is responsible for the looked up operation of routing table. It can dynamically add or delete a routing during running the configuration file, without affecting the normal operation of system. In this paper, we change the operation of modifying routing table into calling a function, and in this function we call a script to modify the routing component in the Click configuration file. When you want to add a routing, then the function of LFB sets the flag to 1, and the script in this function will according to the flag to add the given routing into routing component. When you want to delete a routing, the flag will be set to 0 by this function, and then the script will according to the flag to delete the given routing from routing component. The whole operation without stopping the Click configuration files which are running. In addition to the routing table component, other components such as classification, encapsulation, packet verification, can also be modified by the corresponding LFB.

C. FE virtualization Process Management (FEM) module and Listener module

Here FEM not only can complete the configuration of virtual machine, but also can complete the creation and modification. In addition, you can examine the rest of physical resources on the current FE cluster. Since this platform requires all virtual machines deployed automatically, so it is impossible to manually log on FE cluster to complete the creation of vFE. How to make FE cluster perceive the files transmitted by CEM, and perform immediately after receiving, this is a problem that must be considered when building a platform. To solve this problem, we designed a Listener module, and the module realized by inotify-tools software.

On the FE cluster, we arranged a directory for receiving the creation file of vFE, the Listener monitor the directory through the inotifywait provided by inotify-tools, once the files in the directory are changed, the Listener will inform FEM to take appropriate action. However, what is about the action is to let FEM perform the creation files which have already received, so as to complete the deployment of vFE. When all virtual machines have deployed, FEM will deletes the creation file, and waiting for the next deployment. Listener needs run before deploying the system, and until the system shutdown.



The dotted line boxes indicate modules on CE cluster, the dashed arrow indicates the information interaction between the modules on CE clusters and FE cluster. As it can be seen from the Fig4 the virtual data plane is composed by the following works:

(1) If FEM on FE cluster is responsible for creating vFE, FEM will run the configuration file of virtual machine to inquire about memory, disk size and other parameters of virtual machines. These parameters will be passed to the creation file of virtual machine, then FEM according to the creation file to create a virtual machine.

(2) If CEM on CE cluster is responsible for creating vFE, CEM will send the creation file of virtual machine to FE cluster. When the Listener on FE cluster has monitored this file, and it will inform FEM. At this point FEM executes this file and then accesses OS Automatic Deployment module, The OS Automatic Deployment module is responsible for automatically deploying vCE and vFE, and it is the core module of vForTER automatic platform. This module mainly completes two tasks, on the one hand, the deployment of operating system, on the other hand, the deployment of third party software. Before a real deployment, you need to build the environment for the outer CE, and the environment includes PXE, DHCP, Apache, TFTP, and KickStart.

(3) Once the OS Automatic Deployment module on CE cluster has received the request issued by virtual machine when it had started, then it begins to assign IP addresses for virtual machine. Next it passes the file of booting up system and the file of operating system to virtual machine, so as to complete the deployment of the system. After that, it begins to deploy the environment of third-party software.

(4) Writing the configuration file of Click router, and changing the way of forwarding which used to through the way of calling kernel by LFB, now we use the way of calling Click.

(5) ForCES middleware configurator sets the related configuration file of vFE, and it compiles the environment of software. At this moment, CEM or FEM can add or delete vFE based on the needs of virtual network.

(6) We examine the available physical resources of FE cluster by resource perceptron when deploying vFE, so as to determine whether we could continue to deploy vFE.

D. A resource allocation algorithm for FE based on twice iteration subtraction

When vCE and vFE are deployed successfully, you can configure logic connections and run the ForCES middleware to construct VR. Here the vFE on VR doesn’t require to come from the same FE cluster, and they can be constructed by crossing the FE. You can create various types of virtual network via building the topological structure between VR. Figure 5 shows a virtual network composed of VR.

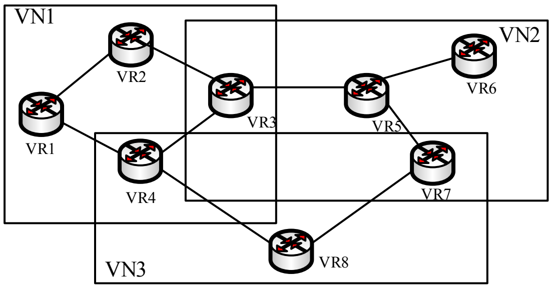


Fig. 5. The virtual network operated in different VR topology

As it can be seen from the figure 5, a VR can be assigned to multiple virtual networks at the same time. The packet processing capability of VR is mainly determined by the set of vFE, the FE device may completely belong to a virtual network. In contrast, it may also be assigned to multiple virtual networks. When FE only contains a vFE, and this vFE fully occupies the physical resources of FE. We say vFE occupies the packet processing capability of FE alone, at this point, this FE only belongs to a virtual network. When the physical resources of FE is not completely occupied by VFE, it is called a non-exclusive occupation, then the FE may belong to a virtual network or multiple virtual networks. Besides FE has the rest of packet processing capability. Typically, FE should not contain too much vFE, that is to say, FE should not be undue virtualization. Packets transfer between excess vFE will bring more extra time overhead. Figure 6 shows the packet size in 512bytes, a transmission rate of 20,000 packets per second, and with the increasing of vFE, the change of the packet processing capability of FE.

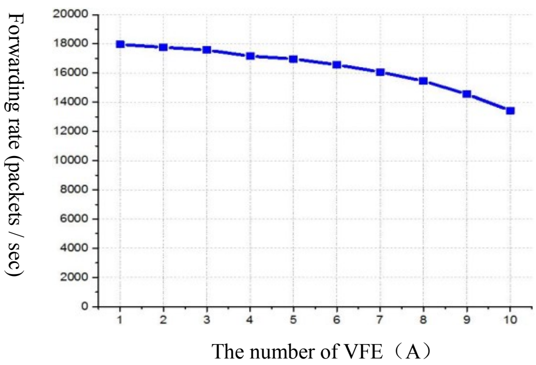


Fig. 6. The relationship between the number of vFE and FE packet processing capacity

As it can be seen from the figure 6, when vFE occupies the packet processing capability of FE alone, the speed of forwarding data will reach maximum. But if there is an increase in the number of vFE, it becomes slower. Therefore, in the virtual network we should try to ensure that vFE occupies FE resources alone, and reduce the degree of FE virtualization appropriately.

In the premise of meeting the requirements of packet processing capacity, when distributing the resource of FE, in order to make the number of FE more reasonable, the use of FE devices more efficient. This paper puts forward an algorithm based on twice iteration subtraction to solve the allocation problem of FE resources. First, we assume that the virtual platform holds the number of virtual network is *M*, and the number of FE available is *n*, we define that every virtual network requires packet processing capacity is V_i ,the upper limit of packet processing capability of FE is a constant C_j , the packet processing capability of FE constitute a collection of $\mathrm{FE_n} = \{ C_1, C_2, \ldots C_n \}$, the packet processing capability of vFE deployed by user is C_t , the upper limit value of *t* depends on the desired packet processing of virtual network. Obviously, the elements in $\mathrm{FE_n}$ must meet the following formula 1:

$$V_i = \sum_{j=1}^{k} C_j + v_i \ (v_i < C_j) \qquad (1)$$

Among them, *K* is a number about the use of FE, v_i is the residual packet processing capacity i-th virtual network required when it has used K-th FE. Here, for each v_i , whose values are smaller than the value C_j , and if you want to make FE reach the upper limit, vFE needs to occupy the resource of FE alone. For a particular vFE and FE, *c=C;* When there is a non-exclusive occupation in FE, *c<C*. Therefore, in this paper, the allocation problem of resources can be abstracted as follows: In the case of satisfying the packet processing capability of i-th virtual network, that is V_i , and give a suitable number of FE which completely occupied by one vFE. In addition, give the number of FE when satisfying the value v_i , It means that satisfying the residual packet processing capacity and there is a non-exclusive occupation in FE.

The process of this algorithm is described as follows:

(1)First, for a given set, we sort the elements in ascending order, and then assign these values to an array S[n]. We define S[n]=$\{ x_1, x_2, \ldots x_n \}$.

(2)Through the array of S[n], and look for a value of x_i , whose value is just equal to the virtual network's packet processing capability *V*. If there is this value, it indicates that the FE which completely occupied by one vFE can meet the conditions. Then just assign these FE to virtual network; if not, then continue to the next step;

(3) Judge the elements in S[n], and make sure that whether these values are higher than *V*; or some of which are higher than *V*, the others are smaller than *V*. If it satisfies the first condition, we only need to find the minimum value x_i=min(s[n]), and then have the corresponding FE been assigned to the virtual network; if it satisfies the second condition, we need to find a value x_i , and meet this condition $x_{i-1} < V$, $x_i > V$, the corresponding FE can meet that demand. If both conditions are not satisfied, then just go on.

(4) At this time, all the elements in S[n] are smaller than *V*. For every element in S[n], they must satisfy the Formula 2.

$$\sum_{j=1}^{k} C_j = a_1 x_1 + a_2 x_2 + \cdots a_n x_n \ (a_i \in \{0, 1\}, \mathrm{i} \in [1, \mathrm{n}]) \qquad (2)$$

Among them, the value of a_i can only be 0 or 1, when the value is 1, it means that assign the FE which corresponds to x_i to the virtual network; when the value is 0, then abandon the corresponding FE. Here, the problem is changed into solving a set of numbers $p=\{ a_1 , a_2 \cdots a_n \}$, and make the value of polynomial $a_1 \ x_1 + a_2 \ x_2 + \cdots a_n \ x_n$ closest to the virtual network's packet processing capability V. Then turn the number set P into a binary sequence Q;

(5)As for Q, there are 2^n-1 kinds of permutations (2^n-1 means removing the case of all zero), and we define that every binary permutation is M, the decimal number which M corresponds to is I, here $1 \leq I \leq 2^n-1$, and I begins to cycle from 1 to 2^n-1, and we have M been stored in the number set P, then make P be brought into formula 2, the calculated result is W.

(6) We define $Z=|W-V|$, if Z=0, the number set P which corresponds to W is required, identify the location of number 1 in P, then assign the corresponding FE to virtual network; if $Z \neq 0$, go on the algorithm.

(7) Record the value Z which obtained in (6), and compare with Z obtained by the next circulation. If the former is larger, then replace the old value with the new value, and enter the next cycle; otherwise go directly into the next round until we find the smallest value Z, and also find the corresponding FE. This is called the first iteration subtraction.

(8)Next, we need to find the most appropriate FE to meet the virtual network v_i , we define the residual packet processing capability of FE is Cu, through the FE which $Cu>V$, and then calculate the minimum value of $Z=Cu-V$ by using the same manner in (7). The FE corresponds to Cu is desired. After that we need to create a vFE in this FE with the package processing capacity is V, and assign it to virtual network. This is the second iteration subtraction.

(9) Synthesize the FE in (6) or (7),(8), that is the optimal solution of FE resource allocation.

V. CONCLUSION

This paper introduces the method of data plane virtualization in detail. In the data plane of this platform, including FE virtual platform, FEM module, the Listener, ForCES middleware configurator and Resource Pool module. This FE virtual platform is based on Click software router as underlying forwarding engine, the FEM module is responsible for the configuration, creation and control of vFE, the Listener monitors the operations of CEM in CE cluster to complete the deployment of vFE by cooperating with CEM and OS automatic deployment device. After the work of data plane virtualization is completed, this paper summarizes the process of the data plane virtualization. Finally, we puts forward an algorithm based on twice iteration subtraction to solve the allocation problem of resources, and to make the number of FE and the utilization of FE reasonable as far as possible.

ACKNOWLEDGMENT

This work was supported in part by a grant from the National Basic Research Program of China (973 Program) (No.2012CB315902), the National Natural Science Foundation of China (No.61402408, 61379120, 61170215), Zhejiang Leading Team of Science and Technology Innovation (No.2011R5001002, 2011R5001003, 2011R5001017). Zhejiang Provincial Key Laboratory of New Network Standards and Technologies (NNST) (No.2013E10012).

REFERENCES

[1] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. Openflow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2):69-74.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Mckeown N, ANDERSON T, PETERSON L, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2):69–74.K. Elissa, "Title of paper if known," unpublished.

[3] DORIA A, SALIM J H, HAAS R, et al.Forwarding and Control Element Separation (ForCES) Protocol Specification[S].RFC 5810(Proposed Standard), Mar. 2010.

[4] KHOSRAVI H, ANDERSON T. Requirements for Separation of IP Control and Forwarding[S]. IETF RFC3654, November 2003.

[5] YANG L, DANTU R, ANDERSON T, et al. Forwarding and Control Element Separation (ForCES) Framework[S]. IETF RFC 3746, April 2004.

[6] HALPERN J, SALIM J H. Forwarding and Control Element Separation (ForCES) Forwarding Element Model[S]. IETF RFC 5812, March 2010.

[7]]WANG W, HALEPLIDIS E, OGAWA K, et al. Forwarding and Control Element Separation (ForCES)Logical Function Block (LFB) Library[S], IETF RFC6956, June 2013.

[8] HAAS R. Forwarding and Control Element Separation (ForCES) MIB[S]. IETF RFC5813, March 2010.

[9] DORIA A, SALIM J H, HAAS R, et al. Forwarding and Control Element Separation (ForCES) Protocol Specification[S]. IETF RFC5810, March 2010.