SDNFV-Based Dynamic Network Function Deployment: Model and Mechanism

Chao Bu, Xingwei Wang, Min Huang, and Keqin Li

***Abstract*— Oriented to the distinctive communication demands of diversified network applications, the current Internet should be able to provide special packet processing operations beyond simple packet forwarding. In this letter, we propose a dynamic network function deployment model based on Software Defined Networking and Network Function Virtualization to control and deploy diverse network functions in corresponding switches, so as to provide special-purpose communication features for different applications. We devise a dynamic network function deployment mechanism, which pre-deploys appropriate functions before they are massively requested according to the prediction, and real-timely deploys a few of new requested functions according to the current network status. The simulation results show that the proposed model is feasible and effective.**

***Index Terms*— Software defined networking, network function virtualization, function deployment, prediction.**

I. Introduction

THE current Internet is successful in supporting data communications for network applications. However, the demands of different applications are becoming distinctive with the rapid development of the Internet technologies. Assume that the network functions can be classified into Basic Network Functions (BNFs) and Special Network Functions (SNFs). The BNFs are necessary for basic packet forwarding (e.g., standard IP routing functions [1]). The SNFs can provide special-purpose communication features for applications, such as security (e.g., firewall), reliability (e.g., error control) and performance enhancement (e.g., traffic shaping). That is, the BNFs play an essential role in forwarding packets while the SNFs play a critical role in satisfying applications specialized demands. Traditionally, the SNFs are usually performed by the intermediary devices [2], which are special hardware based and difficult as well as costly to be managed and deployed in a flexible fashion. It is challenging to control and deploy SNFs flexibly to accommodate the changing demands of different applications.

The Software Defined Networking (SDN) [3] and Network Function Virtualization (NFV) [4] provide an inspiration to deal with the above challenge. SDN decouples network control logic, i.e., control plane, from data plane. NFV decouples network functions from the dedicated network equipment on which they run. An SNF can be dispatched as a plain software based instance, which enables the instantiated SNF to run on the commodity hardware. Thus, based on SDN and NFV (SDNFV), a Dynamic Network Function Deployment (DNFD) model is proposed.

Various SNFs have been developed and are in charge of diverse packet processing operations to adapt to different applications demands. However, it is impossible as well as unnecessary to deploy all SNFs in one single forwarding device (i.e., switch) because of its limited processing capacity. And the method, by which all SNFs are deployed in switches in real time just when they are requested, is unsustainable, especially for the situation that lots of requests for multiple SNFs arrives in a short time slot. It may lead to serious service delay and congestion problem. In fact, some SNFs are frequently used in a switch, while others are hardly used in the switch due to different communication usage patterns which are formed by the combinational influence of environment and time factors. We devise an SNF deployment mechanism, it can pre-deploy appropriate SNFs in switches by predicting the SNFs future popularity according to long-term and short-term prediction. Then, we present an SNF real-time deployment scheme which serves as the supplement to the above pre-deployment one. By them, the SNFs which are very likely to be frequently requested in the next time period can be pre-deployed in switches in advance, and the new requested SNFs can be deployed instantly according to the current network status.

In this letter, we propose a dynamic network function deployment model, the major contributions and innovations can be summarized as follows. Firstly, we devise the SNF deployment mechanism by cooperatively pre-deploying and real-time deploying SNFs, which overcomes the shortcomings (e.g., delay) caused by instantly deploying most of SNFs only when they are requested in, for example, [5] and [6]. Secondly, comparing with just doing prediction for the next time slot [7], the proposed pre-deployment scheme combines long-term and short-term prediction to pre-deploy appropriate SNFs before they are massively requested. Finally, in order to improve the network resource utilization, some constraints (e.g., the switch processing capacity, the link bandwidth) are considered when deploying SNFs.

II. The Proposed DNFD Model

Based on the SDNFV, the system framework of the proposed DNFD is shown in Fig. 1.

The control plane is the decision making center for deploying appropriate SNFs in switches. The Network Function Pool (NFP) contains diverse SNFs which are modularly designed with the standardized interfaces. The Network Function Deployment (NFD) component is charge of making decisions and deploying appropriate SNFs in corresponding switches according to the switch records.

Manuscript received December 29, 2016; accepted January 12, 2017. This work is supported in part by the National Science Foundation for Distinguished Young Scholars of China under Grant 61225012 and Grant 71325002, and in part by the National Natural Science Foundation of China under Grant 61572123. The associate editor coordinating the review of this letter and approving it for publication was V. Eramo. (*Corresponding author: Xingwei Wang*).

C. Bu and X. Wang are with the College of Software, Northeastern University, Shenyang 110819, China (email: bc_0722@126.com; wangxw@mail.neu.edu.cn).

M. Huang is with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (email: mhuang@mail.neu.edu.cn).

K. Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (email: lik@newpaltz.edu).

Digital Object Identifier 10.1109/LCOMM.2017.2654443

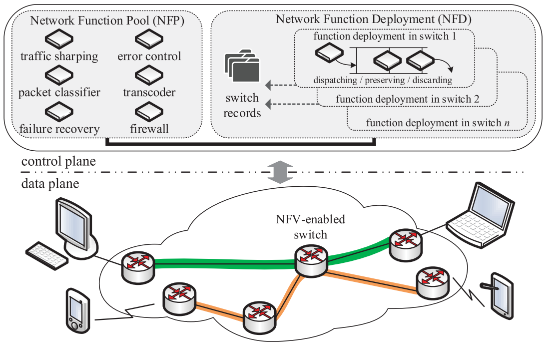

Fig. 1. The system framework of the proposed DNFD model.

The data plane is in charge of supporting communications for network applications. The switches are standardized and NFV-enabled [4], which means that the SNFs can be deployed and removed by the control plane at runtime on the switches. The switches periodically feedback the records of the SNFs which are requested by applications.

III. The Proposed DNFD Mechanism

A. The SNF Popularity

The SNF popularity is defined as how frequently an SNF is requested in a switch. In this letter, we devise a prediction scheme combining long-term prediction and short-term prediction to predict the future Popularity of an SNF to a Switch (PSS).

The long-term prediction is used to determine which SNFs are necessary to a switch for quite a long time. A SNF with high predicted PSS according to the long-term prediction will be preserved in the switch in the next time interval. The long-term prediction is established based on several relatively long time intervals (e.g., a week as a time interval) to avoid the negative influence on such PSSs caused by some special time intervals (e.g., weekends, official holidays). For example, the security related SNFs are always requested in the switches around financial institutions (e.g., bank) during working days, such SNFs predicted PSSs should be based on the long-term prediction rather than the short-term prediction (e.g., daily).

The short-term prediction is mainly oriented to the usually changed SNF requests from applications which may be popular for some relatively short time periods. Thus, the SNFs requested by such applications should be deployed in the corresponding switches in time rather than after the long-term prediction. The short-term prediction is the extension to the long-term prediction.

B. The Mechanism Modelling

1) Notations: Consider the underlying network is denoted as $G = (S, L)$, where S and L are the sets of switches and links respectively. The switch S_l's processing capacity for SNFs is denoted as PCS_l, $S_l \in S$. The link L_e's bandwidth is denoted as BL_e, $L_e \in L$.

Let SNF be the set of SNFs. For $SNF_j \in SNF$, its required processing capacity in a switch is denoted as PC^{SNF_j}. The anti-affinity between SNF_i and SNF_j is denoted as $AAS_{ij} = 1$, which indicates SNF_i and SNF_j cannot be implemented in a same switch, while $AAS_{ij} \neq 1$ indicates the contrary.

Let App be the set of applications. An application functions request is denoted as $\langle App_{id}, S_s^{App_{id}}, S_d^{App_{id}}, SNF^{App_{id}}, BD^{App_{id}} \rangle$, which is assumed according to the NFV-RR [8][9]. App_{id} is the applications identifier, $App_{id} \in App$; $SNF^{App_{id}}$ is the set of the SNFs requested by App_{id} for its special demands, $SNF^{App_{id}} \subseteq SNF$; $BD^{App_{id}}$ is the bandwidth demand of App_{id}; $S_s^{App_{id}}$ and $S_d^{App_{id}}$ are the source switch and destination switch of App_{id} respectively.

Let P^{S_l,S_k} be the set of Z shortest paths between S_l and S_k, $S_l, S_k \in S$, $Z \in N_+$. Each path $P_j^{S_l,S_k}$ is denoted as $\langle S^{P_j^{S_l,S_k}}, L^{P_j^{S_l,S_k}} \rangle$, $S^{P_j^{S_l,S_k}}$ and $L^{P_j^{S_l,S_k}}$ are the sets of switches and links on $P_j^{S_l,S_k}$ respectively, $S^{P_j^{S_l,S_k}} \subseteq S$, $L^{P_j^{S_l,S_k}} \subseteq L$.

2) The Long-Term Prediction: The long-term prediction is mainly for the already deployed SNFs which have been requested in a switch for several time intervals. Assume that $RNS_j^{l,t}$ denotes the requested number of SNF_j in S_l in the $(t)th$ time interval. The actual popularity of SNF_j to S_l in the $(t)th$ time interval is defined as follows:

$$APSS_j^{l,t} = RNS_j^{l,t} \Big/ \sum_{SNF_j \in SNF} RNS_j^{l,t}. \tag{1}$$

The predicted popularity of SNF_j to S_l in the $(t+1)th$ time interval can be obtained according to the actual popularities of SNF_j to S_l in the last t time intervals, $t \in N_+$. The prediction model is defined as follows:

$$APSS_j^{l,t+1} = \sum_{b=1}^{t} \alpha \cdot PPSS_j^{l,b} + \beta. \tag{2}$$

Here, α_1,α_2,..., and α_t are the regression coefficients of $APSS_j^{l,1}$,$APSS_j^{l,2}$,..., and $APSS_j^{l,t}$, β is a constant. We set $A = [\alpha_1, \alpha_2, \ldots, \alpha_t, \beta]$ and its elements can be learned according to the historical popularities of SNF_j to S_l by the typical least squares method defined as follows.

According to the historical records in the switch, the $APSS_j^{l,t}$ is the actual popularity of *SNFj* to *Sl* in the $(x)th$ time interval, we build a matrix M as follows:

$$M = \begin{bmatrix} APSS_j^{l,x} & APSS_j^{l,x+1} & \ldots & APSS_j^{l,x+t} & 1 \\ APSS_j^{l,x+1} & APSS_j^{l,x+2} & \ldots & APSS_j^{l,x+t+1} & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ APSS_j^{l,x+t} & APSS_j^{l,x+t+1} & \ldots & APSS_j^{l,x+2t} & 1 \end{bmatrix}$$
$$= \begin{bmatrix} M_x & M_{x+1} & \ldots & M_{x+t} \end{bmatrix}^T. \tag{3}$$

Here, in the $(x+t+1)th$, $(x+t+2)th$,..., and $(x+2t+1)th$ time intervals, the actual popularities of SNF_j to S_l are $Y = [APSS_j^{l,x+t+1}, APSS_j^{l,x+t+2}, \ldots, APSS_j^{l,x+2t+1}]$, and the predicted popularities of SNF_j to S_l are $[M_x \cdot A^T, M_{x+1} \cdot A^T, \ldots, M_{x+t} \cdot A^T]$ according to (2). We define E_A as follows:

$$E_A = (Y - M \cdot A^T)^T \cdot (Y - M \cdot A^T). \tag{4}$$

The A can be obtained when E_A achieves the minimum:

$$\frac{\partial E_A}{\partial A} = \frac{\partial (Y - M \cdot A^T)^T \cdot (Y - M \cdot A^T)}{\partial A} = 0. \tag{5}$$

In this approach, the predicted PSS of an SNF to a switch in the next time interval can be obtained. We assume $TPSS$ is the threshold to judge the necessity of an SNF to a switch in the next time interval. The set of SNFs which should be in S_l in the $(t+1)th$ time interval is defined as $SNF^{l,t+1}$, and each $SNF_j \in SNF^{l,t+1}$ must satisfy $PPSS_j^{l,t+1} \geq TPSS$.

3) The Short-Term Prediction: In the short-term prediction, we use the daily requested number growth rate of an SNF in a switch to guide deployment of the SNFs which are predicted to be popular in the next day. Such SNFs are deployed in the corresponding switches before the next day. We suppose that the next day of the short-term prediction is one day in the $(t+1)th$ time interval of the long-term prediction.

For SNF_i ($SNF_i \in SNF, SNF_i \notin SNF^{l,t+1}$), its total requested number in S_l in the last m days is defined as $TRNS_i^{l,m}$. The SNF_i's daily incremental requested number in S_l is denoted as follows:

$$DIRNS_i^{l,h} = \begin{cases} TRNS_i^{l,1}, & h = 1 \\ TRNS_i^{l,h} - TRNS_i^{l,h-1}, & 1 < h \leq m. \end{cases} \tag{6}$$

The requested number growth rate of SNF_i in S_l in the $(h)th$ day is denoted as follows:

$$GRS_i^{l,h} = \frac{DIRNS_i^{l,h}}{TRNS_i^{l,m}}, 1 \leq h \leq m. \tag{7}$$

In the short-term prediction, SNF_i is supposed to be increasingly popular in the $(m+1)th$ day as follows:

$$DIRNS_i^{l,m} \geq T \tag{8}$$

$$GRS_i^{l,h} \geq GRS_i^{l,h-1}, \quad 1 \leq h \leq m. \tag{9}$$

Here, $T \in N_+$ is the popularity threshold. We set its predicted popularity label in the $(m+1)th$ day be $PPL_i^{l,m+1} = 1$ for SNF_i which satisfies (8) and (9).

We consider pre-deploying SNFs as many as possible, which are likely to be massively requested (i.e., popular) in each switch before the next day. We define $SNF^{l,t+1}(m+1)$ as the set of SNFs which should be deployed in S_l at the beginning of the $(m+1)th$ day, and $NSNF^{l,t+1}(m+1)$ as the number of its elements. By the followings, we can obtain $SNF^{l,t+1}(m+1)$.

$$maximize \ \frac{SNF^{l,t+1}(m+1)}{\sum_{SNF_i \in SNF^{l,t+1}(m+1)} PC^{SNF_i}} \tag{10}$$

$$s.t. \ \forall SNF_i \in SNF^{l,t+1}(m+1) : PPL_i^{l,m+1} = 1 \tag{11}$$

$$\forall SNF_i \in SNF^{l,t+1}(m+1), \quad \forall SNF_j \in SNF^{l,t+1}$$
$$: AAS_{i,j} \neq 1 \tag{12}$$

$$\forall SNF_i, \forall SNF_j \in SNF^{l,t+1}(m+1),$$
$$SNF_i \neq SNF_j : AAS_{i,j} \neq 1 \tag{13}$$

$$\sum_{SNF_i \in SNF^{l,t+1}(m+1)} PC^{SNF_i} + \sum_{SNF_j \in SNF^{l,t+1}} PC^{SNF_j}$$
$$\leq PCS_l. \tag{14}$$

Thus, at the beginning of the $(m+1)th$ day, the involved SNFs can be dispatched/preserved/discarded to/in/from S_l according to $SNF^{l,t+1} \cup SNF^{l,t+1}(m+1)$, respectively.

4) SNF Real-Time Deployment: In the real time, assume that the set of the currently deployed SNFs in S_l is denoted as $RSNF^l$, the currently occupied processing capacity of S_l is denoted as $RPCS_l$, and the currently occupied bandwidth of L_e is denoted as RBL_e. For an application request $\langle App_i, S_s^{App_i}, S_d^{App_i}, SNF^{App_i}, BD^{App_i} \rangle$, if one of the application App_i's feasible communication paths can satisfy the condition (15):

$$\exists P_j^{S_s^{App_i}, S_d^{App_i}} \in P^{S_s^{App_i}, S_d^{App_i}}, \quad \forall L_e \in L^{P_j^{S_s^{App_i}, S_d^{App_i}}},$$
$$\exists S_l \in S^{P_j^{S_s^{App_i}, S_d^{App_i}}} : BD^{App_i} \leq BL_e - RBL_e,$$
$$SNF^{App_i} \subseteq RSNF^l, \tag{15}$$

it can be directly accepted; otherwise, if one of its feasible communication paths can satisfy the following condition (16):

$$\exists P_j^{S_s^{App_i}, S_d^{App_i}} \in P^{S_s^{App_i}, S_d^{App_i}}, \quad \forall L_e \in L^{P_j^{S_s^{App_i}, S_d^{App_i}}},$$
$$\exists S_l \in S^{P_j^{S_s^{App_i}, S_d^{App_i}}}, \ \forall SNF_a \in SNF^{App_i}, \ \forall SNF_b \in RSNF^l$$
$$: BD^{App_i} \leq BL_e - RBL_e, AAS_{ab} \neq 1,$$
$$\sum_{SNF_k \in (SNF^{App_i} - SNF^{App_i} \cap RSNF^l)} PC^{SNF_k} \leq PCS_l - RPCS_l, \tag{16}$$

the SNFs in $SNF^{App_i} - SNF^{App_i} \cap RSNF^l$ should be deployed instantly in S_l to support App_i until the application finishes. If there are multiple paths satisfying the condition (15) or (16), the path with the least average bandwidth utilization is selected.

In general, a SNF deployed in real time can become a short-term popular SNF or even a long-term popular SNF in a switch if it satisfies the corresponding conditions according to the short-term prediction or the long-term prediction.

IV. Performance Evaluation

In the simulation, we use floodlight as the controller and click modular router as the switch. A variety of SNFs are simulated by ClickOS, which is based on click modular router and can serve as the platform for NFV provisioning. It can create small Virtual Machines (VMs, 6MB), each of which is able to host an SNF. We run each SNF on each ClickOS VM. Considering the network application being a kind of software product, its popularity pattern follows the product life cycle according to the Shifted Gompertz distribution [10]. In this simulation, we classify the application requests on SNFs into 5 types according to 5 usage patterns, each usage pattern is assumed to be mainly oriented to one communication feature, such as security recoverability, stability, high interactivity. For example, Firewall, IPsec, IDS and DPI are security related SNFs. We assume that a switch is mainly oriented to a usage pattern according to its working place. For example, the switches around banks are mainly oriented

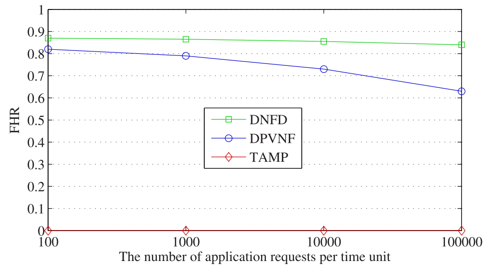

Fig. 2. Function Hit Rate.

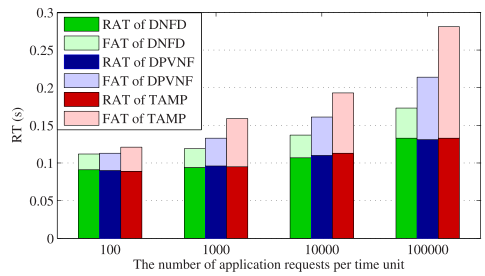

Fig. 3. Response Time Ratio.

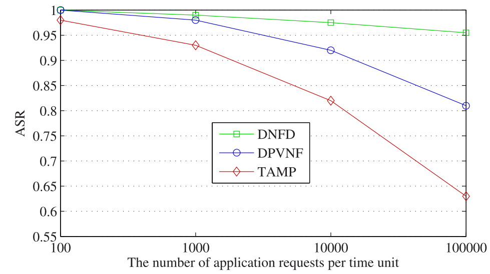

Fig. 4. Access Success Ratio.

to security feature. Each request is assumed to require 1, 2 or 3 related SNFs. For a switch, the application requests are created according to the Shifted Gompertz distribution, the lifetimes of the requests of the switchs usage pattern are set as 200 time units (i.e., long-term popular applications), the lifetimes of the requests of another two selected usage patterns are set as 10 time units (i.e., short-term popular applications), and the lifetimes of the requests of the remaining two usage patterns are set as 1 time units (i.e., occasional applications). We set 7 time units as a time interval for the long-term prediction and 1 time unit for the short-term prediction. The INTERNET2 topology [11] is used for doing simulation, the above 5 usage patterns are randomly and evenly distributed among its nodes. We compare DNFD with Dynamic Placement of Virtual Network Functions (DPVNF) [7] and Traffic-Aware Middlebox Placement (TAMP) [5]. We use the following performance metrics: Function Hit Rate (FHR), Response Time (RT) and Access Success Ratio (ASR).

The FHR is the probability of the requested SNFs having already been deployed. The results are shown in Fig. 2. The DNFD has the highest FHR, followed by DPVNF and TAMP. DNFD and DPVNF can pre-deploy SNFs by prediction, while TAMP just deploys SNFs in real time. DNFD has higher FHR than DPVNF, because DPVNF pre-deploys SNFs mainly based on traffic variation, while DNFD pre-deploys SNFs according to the SNF future popularity which are oriented to SNF demands and thus can effectively improve the FHR.

The RT is the time interval from the application request being received to it being successfully accepted. It contains Request Analysis Time (RAT) and Function Allocation Time (FAT). The results are shown in Fig. 3. DNFD has the lowest RT because its FAT is much shorter than the FAT of DPVNF and TAMP. DNFD and DPVNF can pre-deploy appropriate SNFs by prediction which reduces the FAT, while TAMP just deploys SNFs in real time and then takes much of the RT. DNFD has lower RT than DPVNF, because DNFD can pre-deploy most of the appropriate SNFs by combination of long-term prediction and short-term prediction, while DPVNF just does prediction for the next time slot.

The ASR is the ratio of the number of the successfully accepted application requests to the total number of the received application requests. The results are shown in Fig. 4. DNFD has the highest ASR, followed by DPVNF and TAMP. TAMP just deals with requests and deploys SNFs one by one in real time. When the number of requests increases rapidly, it causes severe service delay by deploying too many SNFs instantly. The ASR of DNFD is higher than that of DPVNF, DNFD not only can pre-deploy more appropriate SNFs in the corresponding switches than DPVNF, but also can real-timely deploys SNFs according to the current network status, which effectively improves the ASR.

V. Conclusions

In this letter, based on SDN and NFV, a dynamic routing function deployment model is proposed to control and deploy diverse SNFs for the specialized communication demands of different applications. Then, an SNF deployment mechanism is devised by long-term prediction and short-term prediction with real-time deployment supplemented, so as to achieve the appropriate SNF deployment in the corresponding switches.

References

[1] F. Baker, "Requirements for IP version 4 routers," document RCF 1812, 1995.

[2] B. Carpenter, "Middleboxes: Taxonomy and issues," document RCF 3234, 2002.

[3] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[4] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2015.

[5] W. Ma, C. Medina, and D. Pan, "Traffic-aware placement of NFV middleboxes," in *Proc. IEEE Globecom*, Dec. 2015, pp. 1–6.

[6] J. Xu, Z. Chen, J. Tang, and S. Su, "T-storm: Traffic-aware online scheduling in storm," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2014, pp. 535–544.

[7] K. Kawashima, T. Otoshi, Y. Ohsita, and M. Murata, "Dynamic placement of virtual network functions based on model predictive control," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2016, pp. 1037–1042.

[8] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, "Optimal network function virtualization realizing end-to-end requests," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[9] "Network function virtualization (NFV): Use cases," document ETSI GS NFV 001, ETSI, Sophia Antipolis, France, 2013.

[10] A. C. Bemmaor, "Modeling the diffusion of new durable goods: Word-of-mouth effect versus consumer heterogeneity," in *Research Traditions in Marketing* (International Series in Quantitative Marketing), vol. 5. Dordrecht, The Netherlands: Springer, 1994, pp. 201–229.

[11] *INTERNET2T*, accessed on Dec. 29, 2016. [Online]. Available: http://www.internet2.edu/