**有人可以解释一下YANG，NETCONF，RESTCONF，XML之间的关系吗？**

[facetothefate](https://www.zhihu.com/people/facetothefate)

[**facetothefate**](https://www.zhihu.com/people/facetothefate)

死理性

87 人赞同了该回答

嗯 简单说一下吧 做要解释这些要从netconf说起。 netconf是什么？ netconf是一种协议用于给网络设备发送配置。 什么意思呢，比如我有一台路由器，我想配一条静态路由，正常的办法是什么呢，我连上面去打命令。

这是最最一般的配置方法了，所以最开始我们网络工程师很吃香，特别是会各种机器设备配置的人。

netconf你可以理解成一种通用的协议，它就是那个会各种配置的人，你只要吩咐他做什么事情，他就会按你说的去做。

这样我的配置就可以写成脚本了，而且关注点变成了配置数据的本身而不是设备的相关命令。

理想的情况，所有的设备都用相同的配置，netconf会自己处理他们到相应的机器的上。听上去很完美是不是？

然而现实是骨干的，网络设备的配置结构往往是是不同的。实现同样的功能的不同设备需要的配置结构也往往不同。比如思科设备接口上配置一个address 只要知道接口名 ip版本 和地址掩码就足够了 但是juniper的机器上不但要知道这些还要额外提供一个unit 号来标识逻辑接口。

这就TM尴尬了。

特别是现在nfv大潮下，nfv是干毛的，网络功能虚拟化，既然虚拟化我就更关注的是功能本身而不是实现这个功能的设备，对于管理人员来说，他往往就只想告诉一套nfv系统，我要什么功能，比如 我要一号站点能联网我要DHCP服务我要VPN，而单纯利用netconf是无法配置的，因为还要求具体的配置结构。这可咋整！

没事儿，我们只要给对应的设备所需的配置结构来个模型不就行了？到时候不就是完形填空吗，yang model就是吃这口饭的，我只需知道对应设备的yang model就可以向管理者请求对应设备所需的信息了，具体结构上的问题有yang model来解释。

什么叫做结构上的问题呢？还是举刚才那个例子

思科为啥不需要unit号来指定逻辑端口呢？因为它的逻辑端口藏在接口名里

所以它的配置类似这样

接口名{

ip{

address x.x.x.x

mask x.x.x.x

}

}

但是juniper的额外需要一个unit号来表示逻辑端口

所以它的配置类似这样

接口名{

unit x {

ip{

address x.x.x.x

mask x.x.x.x

}

}

}

当然实际上他们的yang model 要比这个复杂的多，比如juniper还有family 这个层级这里就是举一个例子

但是对于上层用户管理来说，他只用填逻辑口号，接口名，地址以及ip版本就可以了，他不需要知道具体到底实际的配置长啥样了，因为有可能及其复杂233.

是不是感觉很完美了，这样NFV可以让服务提供商随意换设备了，底层是思科华为还是juniper没啥所谓，反正只要有yang model我们就知道到底怎么配一台机器了。

XML呢，刚说过netconf是一种协议，它实际是个服务器，接受客户端发来的请求来配置自己，所以每台设备上都有一个netconf server，这里的信息交互就是用xml来实现的，所以yang model其实就是一种描述XML结构的模型。

好了，继续向下说到restconf了，这要从rest api说起，需要读者有一点web开发的知识。什么是rest api呢？

说白了这是一种利用HTTP协议来做RPC的办法。我们做NFV呢往往会有一个dashboard，管理员可以利用这个dashboard来配置他想要的功能，这个dashboard很多时候都是一个web application，web application有很多好处，比如客户端可以轻松跨平台，只要有浏览器就行。轻松远程接入等等好处。现代的很多web application往往是这样的结构 一个前端，可以理解为客户端，一个后端，往往就是一个web service了。而REST api就是web service的一种。这里不具体展开了，没有几千字是说不完的。

那么rest api 往往是这样工作的接受一个json格式的数据，然后根据请求的方法和url do something。

我们关注NFV，简单设计一个NFV的dashboard无非就是让用户能去配置机器嘛，

最简单的结构无非就是这样

dashboard

|

| config

|

web service

|

| config + yang

|

netconf client

|

|request XML

|

|

device

这个restconf的作用就在web service 那一层，简单的讲就是怎么把你发来的config和所请求的yang映射到一个URL上的，有了这个URL就能把yang model的数填了，填完了你就有对应的设备的配置结构了，然后根据这个在生成xml就可以把请求发到对应的设备上配置啦。

最后，打个广告，欢迎来试用juniper的contrail哦，开源的SDN controller 基于openstack

[Juniper/contrail-controller](https://link.zhihu.com/?target=https%3A//github.com/Juniper/contrail-controller)

我是PNF的作者之一，希望大家多提issue。（然而我现在已经不维护了所以请用力的提吧！）