# Network Design Report

## Abstract and Overview:

This report will go over the network design of the WDTC ISP network at the Denver site location. The aspects of this design include a fully functioning backbone, external communications with the Level 4 Tier 1 ISP, a highly available and redundant VoIP network, a data center at the Denver location, an automation based network management system, and the ability to serve 10 Tier 3 ISPs. We will be going over the capital and operational expenses of this design as well as the expected timeline for each aspect of the network. Security is our number one priority when it comes to enterprise level networks and these networks include state of the art security implementations that vary from ACLs, to DMZs, firewalls, and much more.

## CapEx/OpEx:

Capital expenses and operational expenses are located in the inventory list.

## Timeline:

Our timeline will be separated into two main parts: hardware and software. The software timeline contains the pre-deployment, deployment and testing and troubleshooting phases. While the hardware timeline will consist of the ordering and installation phase. The timeline is presented below:

**Software:**

**Phase 1 (*Planning/Pre-deployment*):** 8 weeks
**Phase 2 (*Deployment*):** 2 weeks
**Phase 3 (*Testing and Troubleshooting*):** 2 weeks
**Total: 12 weeks (*3 months*)**

**Hardware:**

**Phase 1 (*Ordering equipment)*:** 24 weeks
**Phase 2 (*Installation)*:** 1 week

In total, the expected timeline will take 10 months, adding a couple weeks for unexpected delays.

## IP Space/Allocation:

Facilitating IPs for Tier 3 ISPs involves considering the given number of hosts as per the requirements. The total number of hosts required is 17,503. Providing the “192.168.0.0 / 17” prefix would give us 32,768. Now we will perform subnetting based on the number of hosts per tier 3 site. Eg: For 6498 hosts, we will use 192.168.0.0 / 19 which will give me 8192 (25 % scalable). We have to keep doing this for all the other Tier 3 ISPs.

**Backbone:**

**CLOS Topology:**

For our backbone network, we will be using a clos topology that will have an ingress and egress connection to every other layer of the backbone. Clos allows for almost infinite scalability. It allows for scaling either horizontally or vertically based on the port availability. Also, there are no throughout bottlenecks when considering the clos. We decided on this topology because it reduces the amount of ports used on the networking equipment.

**Devices:**

For the edge routers we settled on the Juniper model with firewall capabilities. Not only can it do up to 100G throughput routing, it can also introduce firewall policies that can restrict traffic based on its source, protocol, and application. It also has built in QoS that can limit the traffic rates to prevent DDOS attacks. We decided on L3 switches for the core and internal devices for high availability when it comes to ports and scalability if more Tier 3 ISPs decide to join our service.

**Routing Protocols:**

For our interior routing protocol, we decided on using OSPF. Compared to its competitors, OSPF has a substantially faster convergence time. For the case of scalability, being able to use route summarization to reduce the amount of routes within the backbone can reduce the amount of routes present in the route table. For the Tier 3 ISP networks, they will connect to the backbone using OSPF, but as a totally stub network. This ensures that any router of any type for future sites can handle OSPF. The main backbone will maintain its own area while each Tier 3 site has their own individual totally stubby area. For any external traffic, we will be using eBGP to get to those destinations.

**Centralized Network Management System:**

The network management system we will be introducing will be hosted in our data center on multiple EC2 instances. There will be VMs running an abstracted web application that allows for easy configuration and deployment. To increase redundancy, multiple VMs will be running this web application so that in the case one fails or goes down, the other will handle its traffic through load balancing. All of the config templates and golden configurations will be stored in S3 buckets as well. To monitor the devices in the backbone, we will be using SNMP to gather metrics that include port status, port utilization, CPU utilization, and uptime. With this data we can create over time graphs that display the amount of utilization and can conclude from there if any changes need to happen. Before configuration changes are put into production, a GNS3 lab environment that mirrors that of the Denver site will implement these changes and need to pass a series of connection based tests before it is passed into the production environment. This GNS3 lab environment will run off of a VM in an EC2 instance. For new automation code, it will have to run successfully in a Jenkins environment and then be implemented into the GNS3 lab

environment before going into production. Change management will happen through Github as golden configs and automation code will be sent once approved through Jenkins and GNS3. IPAM will be updated in the cloud with a database that will insert data based on SNMP information of new devices.

**Tier 3 SDN:**
For the SDN design, the network will consist of a series of open vSwitch SDN capable devices that will route traffic based on the network controller that is an onsite server hosting OpenDayLight as the software. It will begin sending flow tables to the switches via the protocol OpenFlow. Any routing protocols like BGP will be stored in the application layer on the same controller server and the controller will grab those decisions and send them to the data plane.

## Datacenter design

### Requirements

Servers: The infrastructure anticipates 50 servers with an auto-scaling feature for dynamic resource allocation.
Storage Networks (*SAN/NAS*)
High availability and fault-tolerance

### Solution

**AWS**: We opted for AWS as the data center infrastructure due to its cost-efficiency, scalability, and reliability.
**Cost Efficiency**: The solution optimizes costs by minimizing expenses on space, electricity, and maintenance.
**Scalability**: The infrastructure is designed to effortlessly accommodate growing demands.
**Time Efficiency**: The solution prioritizes swift deployment and efficient operations.
**High Availability and Fault Tolerance**: Implemented across two Availability Zones in US West (N. California and Oregon).
**Site-to-Site VPN**: Facilitates secure communication between the backbone network and servers/storage networks.
**EC2 Instance Selection**: We chose t3.medium EC2 instances for their balanced performance in CPU, memory, and network capabilities, making them suitable for a diverse range of workloads.

### Components

- **Amazon Elastic Block Store (Amazon EBS):** Cloud-based block storage service providing persistent and scalable storage volumes for use with Amazon EC2 instances, serving as both SAN (Storage Area Network) and NAS (Network Attached Storage) for diverse data storage requirements.

- **t3.medium EC2 instances**: Server instances offering a balance of memory (4 GiB), network burst bandwidth (*up to 5 Gbps*), and computing power (2 vCPUs), suitable for a wide range of applications with moderate resource requirements.
- **AWS Global Accelerator**: Acts as an edge router, optimizing the global delivery of applications with automatic routing and traffic distribution across AWS regions, while also providing built-in DDoS protection for enhanced security.
- **AWS VPC (*Virtual Private Cloud*)**: Provides a logically isolated section of the AWS Cloud, allowing complete control over network configuration and enabling secure communication between resources, ensuring isolation of network traffic for enhanced security.
- **AWS Transit Gateway:** Serves as a centralized VPN concentrator and gateway for load balancing traffic between multiple regions, simplifying network connectivity and management across diverse environments.
- **Elastic Load Balancer (*ELB*):** Distributes incoming application traffic across multiple targets, such as EC2 instances, within and across availability zones, ensuring high availability by automatically rerouting traffic in case of availability zone failure, thereby enhancing infrastructure resilience and fault tolerance.

**VoIP design**

**Requirements:**
**QoS -** Prioritize voice traffic over others
**Scalability -** Creating Amazon Chimes based on requirements
**Redundancy -** Distribute traffic across different instances in different regions
**Security -** VPC

**Components:**
- **3CX -** 3CX is a software-based private branch exchange (PBX) system that offers unified communications solutions for businesses. This runs on the cloud hosted VMs used for NMAS of our backbone. Used to configure IP Phones and Cloud setup.
- **IP Phones -** Required for internal and external communication using 3CX
- **VPC -** Provides a logically isolated section of the AWS Cloud, allowing complete control over network configuration and enabling secure communication between resources, ensuring isolation of network traffic for enhanced security.
- **AWS Global Accelerator -** Acts as an edge router, optimizing the global delivery of applications with automatic routing and traffic distribution across AWS regions, while also providing built-in DDoS protection for enhanced security.
- **Elastic Load Balancer (*ELB*) -** Distributes incoming application traffic across multiple targets, such as EC2 instances, within and across availability zones, ensuring high

availability by automatically rerouting traffic in case of availability zone failure, thereby enhancing infrastructure resilience and fault tolerance

- **Amazon Chime Voice Connector -** Amazon Chime Voice Connector is a service offered by Amazon Web Services (AWS) that enables businesses to migrate their telephony infrastructure to the cloud. It provides a way to connect on-premises phone systems or private branch exchanges (PBX) with the public switched telephone network (PSTN) using AWS's global network.