

Dheeraj Gajula

dheeraj.gajula@colorado.edu | linkedin.com/in/dheerajgajula/ | github.com/dheerajgajula02 | +1 303-520-8554

EDUCATION

University of Colorado – Boulder

Boulder, CO

Master of Science in Computer Science (Network Engineering) | GPA: 3.8/4.0

Aug 2025 – May 2027

- **Coursework:** Enterprise Networks, Network Systems, Machine Learning, Network Management, SDN, Routing Protocols

Dayananda Sagar College of Engineering

Bangalore, India

Bachelor of Engineering in Computer Science and Engineering | GPA: 3.76/4.0

Nov 2020 – May 2024

- **Coursework:** Data Structures & Algorithms, Database Systems, Operating Systems, Computer Networks, Cloud Computing

EXPERIENCE

Meta

May 2026 – Aug 2026

Production Engineer – Network

Menlo Park, CA

- Built event-driven multi-agent orchestration systems using **MCP tooling** for WhatsApp RCA, reducing oncall triage time by 50%
- Designed performance platform (20+ agents, anomaly detection) monitoring WhatsApp, IG, FB, and CDN with automated reporting
- Engineered MCP abstraction layer bridging network infrastructure and AI systems for deterministic tool-calling workflows

Versa Networks

June 2024 – Aug 2025

Software Engineer – 1

Bangalore, India

- Developed high-concurrency REST APIs in **GoLang (Go)** and **Cassandra** serving 30,000+ reqs/s with **Prometheus** monitoring
- Performed quantitative analysis of VirusTotal feed data using **Python** and **BigQuery**, predicting threat patterns via ML models
- Containerized backend microservices using **Docker** and **Kubernetes (K8s)**, deploying pipelines to **GCP**

Versa Networks

Feb 2024 – June 2024

Software Engineer – Intern

Bangalore, India

- Automated device usage tracking at Versa Networks, reducing billing turnaround time from 7 days to under an hour
- Analyzed inconsistent logs, built automated anomaly detection systems, and implemented device state machines
- Processed director log telemetry using **MongoDB** and Python data modeling, surfacing health metrics on Prometheus
- Built microservices in **Flask** and **FastAPI**, containerizing and deploying services via **Docker Compose**

TECHNICAL PROJECTS

Network Automation Lab | *Python, Ansible, NETCONF, Jenkins, SNMP, Scapy, CI/CD*

Jan 2026 – Present

- Automated BGP and OSPF deployment, validation, and config collection on Cisco routers with Python, Netmiko, and NAPALM
- Built Flask-based tools for automated configuration generation, diff checking, and automated router provisioning
- Developed Ansible playbooks, Jinja2 templates, and NETCONF workflows for repeatable multi-device configuration
- Used SNMP, Nmap, Wireshark, TCPDUMP, and Scapy for real-time network monitoring, packet analysis, and troubleshooting
- Integrated GitHub and Jenkins CI/CD pipelines for automated network configuration linting and testing

Enterprise Network Labs | *STP, VLANs, HSRP, OSPF, EIGRP, RIP, MPLS, RSVP-TE, IPsec, IPv6*

Sept 2025 – Present

- Built fault-tolerant Cisco enterprise topologies with VLANs, trunking, STP/RSTP, and HSRP for default gateway redundancy
- Configured enterprise wireless LANs, DHCP/DHCPv6, inter-VLAN routing, NAT/PAT, and Stateful NAT for Internet connectivity
- Implemented RIP, EIGRP, OSPFv2/v3 multi-area routing, and route redistribution with convergence and failover tuning
- Designed ISP-style topologies with MPLS and RSVP-TE tunnels to support traffic engineering and bandwidth guarantees
- Integrated IPv4/IPv6 dual-stack services with IPsec-encrypted tunnels and NAT-PT for secure cross-protocol translation
- Validated routing, redundancy, and network behavior using Wireshark, traceroute, and Cisco IOS verification tools

Network Applications | *C/C++, Socket Programming, Multi-Threading, Distributed Systems*

Sept 2025 – Present

- Built multithreaded TCP/UDP socket servers, an HTTP proxy, and a custom UDP file transfer protocol in C/C++
- Implemented HTTP parsing, pipelining, persistent connections, cache expiration, and MD5-based page caching
- Enabled concurrent clients using threading/forking, dual-socket architecture, and graceful error recovery routines
- Added URL blocklist filtering, link prefetching, and a secure chat system backed by PostgreSQL and SQLite persistence
- Developed a distributed file system with chunking, redundancy, MD5-based placement, and file retrieval commands

TECHNICAL SKILLS

Protocols: TCP/IP, IPv4, IPv6, BGP, OSPF, MPLS, RSVP-TE, STP/RSTP, DNS, DHCP, NAT/PAT, VPN, IPsec, SDN

Languages & DBs: Python, C/C++, GoLang (Go), Shell Scripting (Bash), SQL (PostgreSQL), Cassandra, InfluxDB, MongoDB

Tools & Cloud: Linux/Unix, Git, Docker, Kubernetes (K8s), Google Cloud Platform (GCP), AWS, NGINX, Prometheus, Grafana

Automation: IaC (Infrastructure as Code), Netmiko, NAPALM, Ansible, Jinja2, NETCONF, Flask, FastAPI, Jenkins, CI/CD

Agentic AI: MCP (Model Context Protocol), Multi-Agent Systems, Agentic Orchestration, Autonomous RCA, Tool Calling