Azure Route Tables & User Defined Routes

Status | Not Started

Document Version: 1.0

Date: January 2026

Audience: Technical Professionals & Network Engineers

Executive Summary: The Postal System Analogy

Think of Azure Route Tables as a city's postal routing system:

Component	Analogy	Purpose
System Routes	Default postal routes (main highways)	Automatic routing within Azure
User Defined Routes	Custom delivery instructions	Force traffic through specific paths
Next Hop Types	Delivery methods (truck, plane, etc.)	Where traffic goes next
Route Tables	Routing rulebook	Collection of routing instructions

Key Insight: UDRs let you override Azure's automatic routing to enforce security, compliance, or custom network paths.

1. Route Tables Fundamentals

1.1 What Are Route Tables?

Technical Definition: Route Tables are collections of routes that determine where network traffic is directed. They control the flow of traffic between subnets, to the internet, and to on-premises networks.

Azure CLI Creation:

bash

```
# Create a basic route table
az network route-table create \
  --name MyRouteTable \
  --resource-group MyResourceGroup \
  --location eastus
```

1.2 System Routes vs User Defined Routes (UDRs)

System Routes (Automatic):

bash

```
# View effective routes on a NIC to see system routes
az network nic show-effective-route-table \
  --name myVMNic \
  --resource-group MyResourceGroup \
  --output table
```

Output shows default system routes:

text

```
Source    State    Address Prefix    Next Hop Type    Next Hop IP
-------   ------   --------------    -------------    -----------
Default   Active   10.0.0.0/16       VnetLocal
Default   Active   0.0.0.0/0         Internet
Default   Active   10.0.0.0/8        None
Default   Active   192.168.0.0/16    None
```

User Defined Routes (Custom):

bash

```
# Create a UDR that overrides default internet route
az network route-table route create \
  --route-table-name MyRouteTable \
  --name ForceToFirewall \
  --resource-group MyResourceGroup \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.1.4
```

1.3 Route Selection Process

Longest Prefix Match Principle:

text

```
Traffic to 10.0.1.10:
1. Route for 10.0.1.10/32 (if exists) → Most specific
2. Route for 10.0.1.0/24
3. Route for 10.0.0.0/16
4. Route for 10.0.0.0/8
5. Route for 0.0.0.0/0 → Least specific
```

Azure CLI to test route selection:

bash

```
# Check effective routes for specific destination
az network watcher test-ip-flow \
  --resource-group MyResourceGroup \
  --direction Outbound \
  --protocol TCP \
  --local 10.0.2.4:443 \
  --remote 8.8.8.8:443 \
  --vm myVM
```

---

2. System Routes Deep Dive

2.1 Default System Routes

Automatic System Routes Table:

yaml

```
System Routes (Cannot be modified):
- 10.0.0.0/16 → VnetLocal (within same VNet)
- 0.0.0.0/0 → Internet (for internet-bound traffic)
- 10.0.0.0/8 → None (Microsoft backbone, blocked)
- 192.168.0.0/16 → None (Microsoft backbone, blocked)
- 100.64.0.0/10 → None (Azure infrastructure)
- GatewaySubnet routes (if VPN/ExpressRoute exists)
```

View all system routes via CLI:

bash

```
# Get effective route table in JSON for detailed view
az network nic show-effective-route-table \
  --name myVMNic \
  --resource-group MyResourceGroup \
  --output json
```

2.2 When System Routes Are Created

Scenario	System Route Added	Purpose
VNet Creation	VnetLocal routes	Intra-VNet communication
Subnet Creation	Subnet-specific routes	Subnet isolation
Public IP Assignment	0.0.0.0/0 → Internet	Internet access
VPN Gateway Creation	On-premises routes	Hybrid connectivity
VNet Peering	Peered VNet routes	Cross-VNet communication

CLI to see route propagation:

bash

```
# Monitor route changes (requires Network Watcher)
az network watcher connection-monitor create \
  --name "RouteMonitor" \
  --resource-group MyResourceGroup \
  --location eastus \
  --source-resource myVM \
  --source-port 80 \
  --dest-address 8.8.8.8 \
  --dest-port 443
```

---

3. User Defined Routes (UDRs) Implementation

3.1 Creating and Managing UDRs

Basic UDR Creation:

bash

```
# Step 1: Create route table
az network route-table create \
  --name CustomRouteTable \
  --resource-group MyResourceGroup \
  --location eastus \
  --disable-bgp-route-propagation false

# Step 2: Add custom routes
az network route-table route create \
  --route-table-name CustomRouteTable \
  --name ToOnPremises \
  --resource-group MyResourceGroup \
  --address-prefix 192.168.1.0/24 \
  --next-hop-type VirtualNetworkGateway

az network route-table route create \
```

```
  --route-table-name CustomRouteTable \
  --name ToInternetViaFW \
  --resource-group MyResourceGroup \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.1.4

# Step 3: Associate with subnet
az network vnet subnet update \
  --name AppSubnet \
  --vnet-name MyVNet \
  --resource-group MyResourceGroup \
  --route-table CustomRouteTable
```

3.2 UDR Next Hop Types

Next Hop Type	CLI Parameter	Use Case	Example
VirtualNetworkGateway	`--next-hop-type VirtualNetworkGateway`	Route to VPN/ExpressRoute	On-premises traffic
VnetLocal	`--next-hop-type VnetLocal`	Force local VNet routing	Intra-VNet optimization
Internet	`--next-hop-type Internet`	Force internet routing	Override default paths
VirtualAppliance	`--next-hop-type VirtualAppliance`	Route through NVA/Firewall	Security inspection
None	`--next-hop-type None`	Blackhole traffic	Security isolation
VnetPeering	`--next-hop-type VNetPeering`	Route to peered VNet	Cross-VNet communication

Example: Blackhole Route for Security:

bash

```
# Create blackhole route for malicious IP ranges
az network route-table route create \
  --route-table-name SecurityRouteTable \
```

```
  --name BlockMaliciousIPs \
  --resource-group MyResourceGroup \
  --address-prefix 185.159.159.0/24 \
  --next-hop-type None
```

3.3 Advanced UDR Scenarios

Scenario 1: Force Tunnel All Internet Traffic

bash

```
# Create route table for force tunneling
az network route-table create \
  --name ForceTunnelRouteTable \
  --resource-group MyResourceGroup \
  --location eastus

# Route all internet traffic to on-premises
az network route-table route create \
  --route-table-name ForceTunnelRouteTable \
  --name ForceToOnPrem \
  --resource-group MyResourceGroup \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualNetworkGateway

# Associate with subnet
az network vnet subnet update \
  --name PrivateSubnet \
  --vnet-name MyVNet \
  --resource-group MyResourceGroup \
  --route-table ForceTunnelRouteTable
```

Scenario 2: Service-Specific Routing

bash

```
# Route specific services through different paths
az network route-table route create \
  --route-table-name ServiceRoutes \
  --name AzureStorageEastUS \
  --resource-group MyResourceGroup \
  --address-prefix 20.150.0.0/16 \
  --next-hop-type Internet

az network route-table route create \
  --route-table-name ServiceRoutes \
  --name AzureSQL \
  --resource-group MyResourceGroup \
  --address-prefix 20.140.0.0/15 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.1.4
```

---

4. Routing Through Azure Firewall/NVA

4.1 Hub-Spoke Architecture with Azure Firewall

Architecture Setup via CLI:

bash

```
# Create Hub VNet
az network vnet create \
  --name HubVNet \
  --resource-group MyResourceGroup \
  --location eastus \
  --address-prefix 10.0.0.0/16 \
  --subnet-name AzureFirewallSubnet \
  --subnet-prefix 10.0.0.0/26

# Create Spoke VNets
az network vnet create \
```

```
  --name Spoke1VNet \
  --resource-group MyResourceGroup \
  --location eastus \
  --address-prefix 10.1.0.0/16

az network vnet create \
  --name Spoke2VNet \
  --resource-group MyResourceGroup \
  --location eastus \
  --address-prefix 10.2.0.0/16

# Deploy Azure Firewall
az network firewall create \
  --name HubFirewall \
  --resource-group MyResourceGroup \
  --location eastus \
  --vnet-name HubVNet \
  --public-ip-count 1

# Get Firewall Private IP
FW_PRIVATE_IP=$(az network firewall show \
  --name HubFirewall \
  --resource-group MyResourceGroup \
  --query "ipConfigurations[0].privateIpAddress" \
  --output tsv)

# Create Route Table for Spokes
az network route-table create \
  --name SpokeRouteTable \
  --resource-group MyResourceGroup \
  --location eastus

# Route all traffic through Firewall
az network route-table route create \
  --route-table-name SpokeRouteTable \
  --name DefaultRoute \
```

```
  --resource-group MyResourceGroup \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address $FW_PRIVATE_IP

# Associate with spoke subnets
az network vnet subnet update \
  --name AppSubnet \
  --vnet-name Spoke1VNet \
  --resource-group MyResourceGroup \
  --route-table SpokeRouteTable
```

4.2 Network Virtual Appliance (NVA) Routing

Deploying and Configuring NVA:

bash

```
# Deploy NVA VM (Example: Cisco CSR 1000V)
az vm create \
  --name NVA-VM \
  --resource-group MyResourceGroup \
  --location eastus \
  --image cisco:cisco-csr-1000v:16_12_4-byol:latest \
  --admin-username azureuser \
  --admin-password "SecurePassword123!" \
  --vnet-name HubVNet \
  --subnet NVA-Subnet \
  --public-ip-address "" \
  --nsg ""

# Enable IP Forwarding on NVA NIC
az network nic update \
  --name NVA-VM-nic \
  --resource-group MyResourceGroup \
  --ip-forwarding true
```

```
# Create UDRs to route through NVA
az network route-table route create \
  --route-table-name NVARoutes \
  --name ToInternetViaNVA \
  --resource-group MyResourceGroup \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.2.4  # NVA Private IP

az network route-table route create \
  --route-table-name NVARoutes \
  --name ToOnPremViaNVA \
  --resource-group MyResourceGroup \
  --address-prefix 192.168.0.0/16 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.2.4
```

4.3 Asymmetric Routing Prevention

Problem: Traffic goes through firewall/NVA in one direction but returns via different path.

Solution:

bash

```
# Ensure return path is also through firewall
# Create UDRs on ALL subnets/gateways

# On GatewaySubnet (if using VPN/ExpressRoute)
az network route-table route create \
  --route-table-name GatewayRoutes \
  --name ToSpokesViaFirewall \
  --resource-group MyResourceGroup \
  --address-prefix 10.1.0.0/16 \
  --next-hop-type VirtualAppliance \
```

```
  --next-hop-ip-address $FW_PRIVATE_IP

az network route-table route create \
  --route-table-name GatewayRoutes \
  --name ToSpokesViaFirewall2 \
  --resource-group MyResourceGroup \
  --address-prefix 10.2.0.0/16 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address $FW_PRIVATE_IP
```

---

5. Controlling Routing Behavior

5.1 Inbound vs Outbound Routing

Inbound Traffic Control:

bash

```
# Route inbound internet traffic through WAF
az network route-table route create \
  --route-table-name InboundRoutes \
  --name InternetToWeb \
  --resource-group MyResourceGroup \
  --address-prefix 203.0.113.0/24  # Your public IP range
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.3.4   # WAF Private IP
```

Outbound Traffic Control:

bash

```
# Route outbound traffic through proxy
az network route-table route create \
  --route-table-name OutboundRoutes \
  --name WebToInternet \
  --resource-group MyResourceGroup \
```

```
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.4.4   # Proxy Server IP
```

5.2 BGP Route Propagation

Enable/Disable BGP Routes:

bash

```
# Create route table without BGP propagation
az network route-table create \
  --name NoBGPRoutes \
  --resource-group MyResourceGroup \
  --location eastus \
  --disable-bgp-route-propagation true

# Create route table with BGP propagation
az network route-table create \
  --name WithBGPRoutes \
  --resource-group MyResourceGroup \
  --location eastus \
  --disable-bgp-route-propagation false
```

View BGP Learned Routes:

bash

```
# Check BGP learned routes on ExpressRoute circuit
az network express-route peering route-table summary list \
  --resource-group MyResourceGroup \
  --circuit-name MyExpressRoute \
  --peering-name AzurePrivatePeering
```

5.3 Route Prioritization and Overrides

Route Priority (Longest Prefix Match):

bash

```
# More specific routes override less specific ones
az network route-table route create \
  --route-table-name PriorityRoutes \
  --name SpecificRoute \
  --resource-group MyResourceGroup \
  --address-prefix 10.0.1.0/24 \      # More specific
  --next-hop-type Internet

az network route-table route create \
  --route-table-name PriorityRoutes \
  --name GeneralRoute \
  --resource-group MyResourceGroup \
  --address-prefix 10.0.0.0/16 \      # Less specific
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.1.4
```

Force Override with /32 Routes:

bash

```
# Route specific IP through different path
az network route-table route create \
  --route-table-name OverrideRoutes \
  --name OverrideSpecificVM \
  --resource-group MyResourceGroup \
  --address-prefix 10.0.2.10/32 \     # Most specific
  --next-hop-type None                # Block this VM
```

---

6. Architecture Diagrams & Implementations

6.1 Hub-Spoke with Forced Tunneling

CLI Implementation:

bash

```
#!/bin/bash
# Complete Hub-Spoke with Force Tunneling Setup

# Variables
RG="Network-RG"
LOCATION="eastus"
HUB_VNET="Hub-VNet"
SPOKE_VNET="Spoke-VNet"
FIREWALL_NAME="Hub-Firewall"

# Create Resource Group
az group create --name $RG --location $LOCATION

# Create Hub VNet with GatewaySubnet
az network vnet create \
  --name $HUB_VNET \
  --resource-group $RG \
  --address-prefix 10.0.0.0/16 \
  --subnet-name GatewaySubnet \
  --subnet-prefix 10.0.0.0/27

# Create AzureFirewallSubnet (must be exactly /26)
az network vnet subnet create \
  --name AzureFirewallSubnet \
  --vnet-name $HUB_VNET \
  --resource-group $RG \
  --address-prefix 10.0.1.0/26

# Create Spoke VNet
az network vnet create \
  --name $SPOKE_VNET \
  --resource-group $RG \
  --address-prefix 10.1.0.0/16 \
  --subnet-name WorkloadSubnet \
```

```
  --subnet-prefix 10.1.0.0/24

# Deploy Azure Firewall
az network firewall create \
  --name $FIREWALL_NAME \
  --resource-group $RG \
  --location $LOCATION \
  --vnet-name $HUB_VNET \
  --public-ip-count 1

# Get Firewall Private IP
FW_IP=$(az network firewall show \
  --name $FIREWALL_NAME \
  --resource-group $RG \
  --query "ipConfigurations[0].privateIpAddress" \
  --output tsv)

# Create Route Table for Spoke
az network route-table create \
  --name Spoke-RouteTable \
  --resource-group $RG \
  --location $LOCATION

# Force all traffic through Firewall
az network route-table route create \
  --route-table-name Spoke-RouteTable \
  --name ToInternetViaFirewall \
  --resource-group $RG \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address $FW_IP

# Associate Route Table with Spoke Subnet
az network vnet subnet update \
  --name WorkloadSubnet \
  --vnet-name $SPOKE_VNET \
```

```
  --resource-group $RG \
  --route-table Spoke-RouteTable

echo "Setup complete. Firewall IP: $FW_IP"
```

6.2 Multi-Tier Application Routing

Architecture Diagram via Mermaid (for documentation):

CLI Implementation:

bash

```
# Create route tables for each tier
az network route-table create \
  --name WebTier-Routes \
  --resource-group $RG \
  --location $LOCATION

az network route-table create \
  --name AppTier-Routes \
  --resource-group $RG \
  --location $LOCATION

az network route-table create \
  --name DataTier-Routes \
  --resource-group $RG \
  --location $LOCATION

# Web Tier: Direct internet access
az network route-table route create \
  --route-table-name WebTier-Routes \
  --name ToInternet \
  --resource-group $RG \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type Internet
```

```
# App Tier: Route through Firewall to internet
az network route-table route create \
  --route-table-name AppTier-Routes \
  --name ToInternetViaFW \
  --resource-group $RG \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address $FW_IP

# Data Tier: No internet access, only local
az network route-table route create \
  --route-table-name DataTier-Routes \
  --name ToAppTier \
  --resource-group $RG \
  --address-prefix 10.0.2.0/24 \
  --next-hop-type VnetLocal

az network route-table route create \
  --route-table-name DataTier-Routes \
  --name BlockInternet \
  --resource-group $RG \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type None  # Blackhole
```

---

7. Advanced Routing Scenarios

7.1 Service Tag Based Routing

Route Azure Services through Firewall:

bash

```
# Get Azure Service Tag IP ranges
az network list-service-tags --location eastus --output table

# Create routes for specific service tags
```

```
az network route-table route create \
  --route-table-name ServiceRoutes \
  --name AzureStorage \
  --resource-group MyResourceGroup \
  --address-prefix 20.150.0.0/16 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.1.4

az network route-table route create \
  --route-table-name ServiceRoutes \
  --name AzureSQL \
  --resource-group MyResourceGroup \
  --address-prefix 20.140.0.0/15 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.1.4
```

7.2 Geographic Routing

Route traffic based on destination geography:

bash

```
# Route European traffic through different path
az network route-table route create \
  --route-table-name GeoRoutes \
  --name EuropeTraffic \
  --resource-group MyResourceGroup \
  --address-prefix 91.198.0.0/16 \
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.5.4  # EU Proxy

# Route US traffic directly
az network route-table route create \
  --route-table-name GeoRoutes \
  --name USTraffic \
  --resource-group MyResourceGroup \
```

```
  --address-prefix 8.0.0.0/8 \
  --next-hop-type Internet
```

7.3 Dynamic Routing with Automation

Automated Route Management Script:

bash

```
#!/bin/bash
# Dynamic route management based on threat intelligence

RG="Security-RG"
RT_NAME="ThreatIntel-Routes"
THREAT_FEED_URL="https://threatfeeds.azure.com/malicious-ips.json"

# Fetch threat intelligence
curl -s $THREAT_FEED_URL | jq -r '.ip_ranges[]' > malicious_ips.txt

# Create or update route table
az network route-table create \
  --name $RT_NAME \
  --resource-group $RG \
  --location eastus \
  --output none

# Add routes for malicious IPs
while read IP_RANGE; do
  az network route-table route create \
    --route-table-name $RT_NAME \
    --name "Block_$(echo $IP_RANGE | tr './' '_')" \
    --resource-group $RG \
    --address-prefix $IP_RANGE \
    --next-hop-type None
done < malicious_ips.txt
```

```
echo "Threat intelligence routes updated"
```

---

8. Monitoring & Troubleshooting

8.1 Route Monitoring Commands

Check Effective Routes:

bash

```
# View effective routes for a VM
az network nic show-effective-route-table \
  --name myVMNic \
  --resource-group MyResourceGroup \
  --output table

# Check specific route
az network route-table route show \
  --name MyRoute \
  --route-table-name MyRouteTable \
  --resource-group MyResourceGroup
```

Monitor Route Changes:

bash

```
# Enable Diagnostic Settings for Route Table
az monitor diagnostic-settings create \
  --resource $(az network route-table show \
    --name MyRouteTable \
    --resource-group MyResourceGroup \
    --query id -o tsv) \
  --name RouteTableDiagnostics \
  --storage-account $(az storage account show \
    --name mystorageaccount \
```

```
    --resource-group MyResourceGroup \
    --query id -o tsv) \
  --logs '[{"category": "NetworkSecurityGroupRuleCounter", "enabled": true}]'
```

8.2 Common Troubleshooting Scenarios

Issue 1: Route Not Taking Effect

bash

```
# Check route table association
az network vnet subnet show \
  --name MySubnet \
  --vnet-name MyVNet \
  --resource-group MyResourceGroup \
  --query routeTable.id

# Check effective routes
az network nic show-effective-route-table \
  --name $(az vm show \
    --name MyVM \
    --resource-group MyResourceGroup \
    --query 'networkProfile.networkInterfaces[0].id' \
    --output tsv | awk -F/ '{print $NF}') \
  --resource-group MyResourceGroup \
  --output table
```

Issue 2: Asymmetric Routing

bash

```
# Check both directions
az network watcher test-ip-flow \
  --resource-group MyResourceGroup \
  --direction Outbound \
  --protocol TCP \
  --local 10.0.2.4:443 \
```

```
  --remote 8.8.8.8:443 \
  --vm myVM

az network watcher test-ip-flow \
  --resource-group MyResourceGroup \
  --direction Inbound \
  --protocol TCP \
  --local 10.0.2.4:443 \
  --remote 8.8.8.8:443 \
  --vm myVM
```

Issue 3: BGP Routes Not Propagating

bash

```
# Check BGP status
az network express-route peering list \
  --resource-group MyResourceGroup \
  --circuit-name MyExpressRoute

# Check route propagation settings
az network route-table show \
  --name MyRouteTable \
  --resource-group MyResourceGroup \
  --query "disableBgpRoutePropagation"
```

8.3 Diagnostic Tools

Network Watcher for Routing:

bash

```
# Install Network Watcher extension
az extension add --name network-watcher

# Test connectivity with route tracing
az network watcher test-connectivity \
```

```
  --resource-group MyResourceGroup \
  --source-resource mySourceVM \
  --dest-resource myDestVM \
  --dest-port 80

# Get next hop information
az network watcher show-next-hop \
  --resource-group MyResourceGroup \
  --vm myVM \
  --source-ip 10.0.1.4 \
  --destination-ip 8.8.8.8
```

---

9. Security Best Practices

9.1 Route Table Security

Implement Least Privilege Routing:

bash

```
# Create minimal required routes only
az network route-table route create \
  --route-table-name MinimalRoutes \
  --name OnlyRequired \
  --resource-group MyResourceGroup \
  --address-prefix 10.0.0.0/16 \
  --next-hop-type VnetLocal

# Deny all other traffic
az network route-table route create \
  --route-table-name MinimalRoutes \
  --name DenyAll \
  --resource-group MyResourceGroup \
  --address-prefix 0.0.0.0/0 \
  --next-hop-type None
```

Audit Route Tables:

bash

```
# List all route tables with routes
az network route-table list \
  --resource-group MyResourceGroup \
  --query "[].{Name:name, Routes:routes[].name}" \
  --output table

# Check for overly permissive routes
az network route-table route list \
  --route-table-name MyRouteTable \
  --resource-group MyResourceGroup \
  --query "[?addressPrefix=='0.0.0.0/0']" \
  --output table
```

9.2 Compliance and Governance

Azure Policy for Route Tables:

bash

```
# Assign policy to enforce routing through firewall
az policy assignment create \
  --name "enforce-firewall-routing" \
  --display-name "Enforce Firewall Routing" \
  --policy "/providers/Microsoft.Authorization/policyDefinitions/.../enforce-rou
te-through-firewall" \
  --params '{"effect": "deny"}' \
  --scope "/subscriptions/{subscription-id}"
```

Tagging for Organization:

bash

```
# Tag route tables for cost management
az network route-table update \
```

```
  --name MyRouteTable \
  --resource-group MyResourceGroup \
  --set tags.Environment=Production tags.CostCenter=IT
```

---

10. Performance Optimization

10.1 Route Table Limits and Scaling

Azure Limits:

bash

```
# Check current usage
az network route-table list \
  --resource-group MyResourceGroup \
  --query "[].{Name:name, Routes:length(routes)}" \
  --output table

# Limits:
# - 200 routes per route table
# - 200 route tables per subscription (can be increased)
# - 1 route table per subnet
```

Optimization Strategies:

bash

```
# Combine routes using larger prefixes when possible
az network route-table route create \
  --route-table-name OptimizedRoutes \
  --name CombineNetworks \
  --resource-group MyResourceGroup \
  --address-prefix 10.0.0.0/16 \  # Instead of multiple /24s
  --next-hop-type VirtualAppliance \
  --next-hop-ip-address 10.0.1.4
```

10.2 Route Propagation Delay

Minimize Propagation Time:

bash

```
# Use Azure Firewall for faster propagation vs custom NVA
# Firewall routes propagate immediately
# Custom NVA may have delays

# Monitor propagation
az network watcher connection-monitor create \
  --name "RoutePropagationMonitor" \
  --resource-group MyResourceGroup \
  --location eastus \
  --source-resource myVM \
  --source-port 80 \
  --dest-address 8.8.8.8 \
  --dest-port 443 \
  --monitoring-interval 30
```

---

11. Complete Deployment Example

11.1 Enterprise Network Deployment Script

bash

```
#!/bin/bash
# Complete Enterprise Network with Advanced Routing

set -e  # Exit on error

# Configuration
RG="Enterprise-Network-RG"
LOCATION="eastus"
HUB_VNET="Enterprise-Hub"
```

```
SPOKES=("Finance" "HR" "Engineering")
FIREWALL_NAME="Enterprise-FW"

echo "=== Creating Enterprise Network ==="

# Create Resource Group
az group create --name $RG --location $LOCATION

# Create Hub VNet
az network vnet create \
  --name $HUB_VNET \
  --resource-group $RG \
  --address-prefix 10.0.0.0/22 \
  --subnet-name AzureFirewallSubnet \
  --subnet-prefix 10.0.0.0/26

# Deploy Azure Firewall
echo "Deploying Azure Firewall..."
az network firewall create \
  --name $FIREWALL_NAME \
  --resource-group $RG \
  --location $LOCATION \
  --vnet-name $HUB_VNET \
  --public-ip-count 2 \
  --sku-name AZFW_VNet \
  --sku-tier Standard

FW_IP=$(az network firewall show \
  --name $FIREWALL_NAME \
  --resource-group $RG \
  --query "ipConfigurations[0].privateIpAddress" \
  --output tsv)

# Create Spoke Networks
for SPOKE in "${SPOKES[@]}"; do
  echo "Creating Spoke: $SPOKE"
```

```
# Create VNet
az network vnet create \
  --name "Spoke-$SPOKE" \
  --resource-group $RG \
  --address-prefix "10.1.$((RANDOM % 250)).0/24"

# Create Subnets
az network vnet subnet create \
  --name "Web-$SPOKE" \
  --vnet-name "Spoke-$SPOKE" \
  --resource-group $RG \
  --address-prefix "10.1.$((RANDOM % 250)).0/26"

az network vnet subnet create \
  --name "App-$SPOKE" \
  --vnet-name "Spoke-$SPOKE" \
  --resource-group $RG \
  --address-prefix "10.1.$((RANDOM % 250)).64/26"

az network vnet subnet create \
  --name "Data-$SPOKE" \
  --vnet-name "Spoke-$SPOKE" \
  --resource-group $RG \
  --address-prefix "10.1.$((RANDOM % 250)).128/26"

# Create Route Tables for each tier
for TIER in Web App Data; do
  RT_NAME="$TIER-$SPOKE-Routes"

  az network route-table create \
    --name $RT_NAME \
    --resource-group $RG \
    --location $LOCATION

  # Different routes per tier
```

```
case $TIER in
  Web)
    # Web tier can access internet directly
    az network route-table route create \
      --route-table-name $RT_NAME \
      --name ToInternet \
      --resource-group $RG \
      --address-prefix 0.0.0.0/0 \
      --next-hop-type Internet
    ;;
  App)
    # App tier through firewall
    az network route-table route create \
      --route-table-name $RT_NAME \
      --name ToInternetViaFW \
      --resource-group $RG \
      --address-prefix 0.0.0.0/0 \
      --next-hop-type VirtualAppliance \
      --next-hop-ip-address $FW_IP
    ;;
  Data)
    # Data tier no internet
    az network route-table route create \
      --route-table-name $RT_NAME \
      --name NoInternet \
      --resource-group $RG \
      --address-prefix 0.0.0.0/0 \
      --next-hop-type None
    ;;
esac

# Associate route table
az network vnet subnet update \
  --name "$TIER-$SPOKE" \
  --vnet-name "Spoke-$SPOKE" \
  --resource-group $RG \
```

```
    --route-table $RT_NAME
done

# Peer with Hub
az network vnet peering create \
  --name "HubTo$SPOKE" \
  --resource-group $RG
```