DASHSOFT
CONCEPT - CODE - CLOUD

Datum: 26.11.2025

Name: Automatically Inserted

Incident Report: Network Subsystem Failure

Compiled by: Julian R.Date: 2025-01-22

---

1. Executive Summary

On 21 January 2025, the internal compute cluster at Novelized experienced a multi-stage network interruption that led to degraded API performance, delayed batch jobs, and a brief loss of access to the administration panel.

This document outlines:

- *The timeline of the incident*
- *Technical details*
- *Root-cause analysis*
- *Mitigation and recommendations*
- *Post-incident review*

---

2. Incident Overview

The disruption began at approximately 14:03 CET when the monitoring agent detected an unexpected spike in packet loss across Node-07.Shortly after, multiple services escalated error rates beyond standard thresholds.

The following symptoms were confirmed:

- Elevated request latency
- Partial API unavailability
- Unreachable internal endpoints
- Failed database connection attempts
- Degraded performance in distributed schedulers

---

3. Affected Systems

- Cluster Node-07 (storage + compute)
- Node-03 (gateway)
- The internal LB subsystem
- Redis cache fabric
- The message queue dispatcher
- Internal developer portal (“Toolbelt”)

---

4. Timeline of Events

4.1 Detection Phase

14:03 – Packet loss spike detected.14:04 – Health monitor reports intermittent heartbeat failure.14:06 – Alert propagated to on-call engineer.14:08 – Gateway begins dropping internal service discovery packets.14:11 – Developer portal becomes unreachable.14:15 – Distributed scheduler reports missing worker heartbeats.

4.2 Investigation Phase

14:17 – On-call begins remote diagnostic on Node-07.14:18 – Traceroute reveals inconsistent hops within internal WAN.14:19 – Logs indicate collapsed routing table on gateway Node-03.14:21 – Attempts to restore routing table fail.14:23 – Secondary analysis suggests ARP table contamination.14:24 – Node-07 network driver emits repeated warnings.14:27 – Multiple nodes begin corrective reconnection attempts.

4.3 Intervention Phase

14:30 – Engineering triggers rolling failover.14:33 – Gateway Node-03 temporarily taken offline.14:34 – Router cache flushed.14:36 – ARP tables manually corrected.14:38 – Nodes rejoin the cluster.14:40 – Internal LB resets epoch timestamps.14:42 – All services begin stabilizing.

4.4 Resolution Phase

14:50 – API error rates return to normal.14:52 – Scheduler resumes queued tasks.14:54 – All internal services operational.15:00 – Incident declared resolved.

---

5. Technical Deep Dive

5.1 Observed Error Patterns

- Repeated ARP lookup failures
- Gateway discarding broadcast packets
- Sudden reduction in available network interfaces
- Unstable link-state transitions
- Unexpected “device busy” kernel messages

5.2 Initial Hypotheses

- Hypothesis A: Corrupted routing table due to improper shutdown
- Hypothesis B: Network driver regression introduced in previous patch
- Hypothesis C: Misconfigured VLAN tagging from Node-03
- Hypothesis D: Hardware-level NIC degradation

Only Hypothesis C and Hypothesis D showed high alignment with observed symptoms.

5.3 Verification

- VLAN tagging rules examined manually
- Packet captures from Node-07 compared to control node
- NIC health audit performed
- Recent kernel patches verified
- Neighbor discovery patterns examined

---

6. Root Cause Analysis

6.1 Primary Cause

The immediate trigger was a faulty NIC on Node-07, which intermittently misreported its MAC address. This caused circulating ARP announcements with contradictory identity claims.

6.2 Secondary Factors

- Misconfigured fallback VLAN rules amplified routing confusion
- Gateway Node-03 did not throttle ARP announcements correctly
- Monitoring agent lacked a specific check for ARP churn
- Maintenance window the day before introduced unusual timing conditions

6.3 Why the Issue Escalated

- ARP poisoning-like behavior flooded the gateway
- Internal LB lost track of correct backend interfaces
- Multiple nodes attempted to self-correct, creating more traffic
- Cluster health scoring triggered cascading downgrades

---

7. Logs & Observations

7.1 Extracted Network Driver Messages

- “interface reports inconsistent hardware address”
- “arp cache overflow detected”
- “device reinitializing link state”
- “attempted to allocate stale descriptor”

7.2 Behavioral Traits

- Recurring bursts every 30–45 seconds
- CPU spikes coinciding with ARP storms
- Rapid succession of “neighbor changed” messages
- Unusual time-to-live mismatches in internal requests

---

8. Actions Taken During Incident

- Manual override of route resolution
- Clearing gateway routing tables
- Restarting NIC driver on Node-07
- Forcing cluster domain rebalance
- Disabling problematic fallback VLAN rule
- Rolling refresh of LB and internal cache fabric

---

9. Interviews and Notes

9.1 Interview: On-Call Engineer

- “I observed unpredictable ARP churn that did not match prior patterns.”
- “The gateway did not respond to standard command sequences.”
- “There was a brief period where logs contradicted each other.”

9.2 Interview: Network Lead

- *“The broadcast storm likely stemmed from partial NIC failure.”*
- *“Our fallback VLAN configuration lacked redundancy.”*
- *“Mitigation time is acceptable but can be reduced.”*

---

10. Resolution & Recovery

10.1 Immediate Fix

- *NIC on Node-07 replaced*
- *Gateway recommissioned*
- *ARP storm subsided*

10.2 Post-Recovery Verification

- *6-hour stability monitoring*
- *Traffic patterns consistent with baseline*
- *All heartbeats restored*
- *Scheduler throughput returned to normal*

---

11. Recommendations

- *Introduce ARP churn alert level*
- *Add NIC health checks to daily tests*
- *Harden gateway fallback VLAN configuration*
- *Add redundancy to routing table commits*
- *Improve documentation for network failover paths*
- *Add periodic driver stress tests*
- *Maintain a record of interface identity history*

---

12. Lessons Learned

- *Hardware degradation can mimic malware behavior*
- *Conflicting identity signals cause systemic confusion*
- *VLAN misconfigurations scale their damage quickly*
- *Monitoring should detect patterns, not just thresholds*
- *Recovery scripts must handle ARP floods more gracefully*

---

13. Appendix A: Secondary Notes

This section includes non-critical but relevant observations:

- *Engineers detected only slight warming on the faulty NIC*
- *ARP rate exceeded 40× typical levels*
- *Routing cache contained 112 invalid entries at one point*
- *Link state transitions correlated with scheduler bursts*

---

14. Appendix B: Service Impact Overview

- *API latency increased by 204%*
- *Developer portal inaccessible for 18 minutes*
- *Queue processing delayed by 32 minutes*
- *Peak error rate: 17.3%*

---

15. Final Statement

The incident demonstrates how fragile trust relationships are within distributed systems when even a single component begins emitting misleading signals.

A healthy network is less about bandwidth and more about identity, consensus, and stability.