FX3110 Ethernet WAN Failover Test Report

Date: 2026-01-12 **Product**: Inseego FX3110 5G Indoor Router **Issue**: Ethernet WAN does not failover to Cellular when upstream internet is blocked **Reported by**: Geoffrey Noakes

---

Executive Summary

The Inseego FX3110 fails to automatically failover from Ethernet WAN to Cellular WAN when upstream internet connectivity is lost, despite the Ethernet link remaining physically active. This creates a critical failure scenario where end devices (POS terminals, Raspberry Pi, etc.) lose internet connectivity even though the FX3110 has a functional cellular backup connection available.

Expected Behavior: When end devices cannot reach the internet through Ethernet WAN, the FX3110 should detect the failure and switch to Cellular WAN.

Actual Behavior: The FX3110 remains on Ethernet WAN as long as the physical link is up and DHCP is functional, even when internet connectivity is completely blocked upstream.

Network Topology

Test Environment Architecture

```
      INTERNET
         |
         |

 Google Nest WiFi
 Router/Gateway
 192.168.86.1



BLOCKED/FILTERED         <-- Traffic blocked here
(Device blocking)



Inseego FX3110
Ethernet WAN Port
WAN IP: 192.168.86.48
```

```
                           Status: "Connected"


                                eth0 192.168.9.1 (FX3110 LAN)

                           Raspberry Pi
                           eth0: 192.168.9.14
                           wlan0: 192.168.86.38
```

FX3110 Configuration Details

- **Ethernet WAN (Primary)**:
  - Interface: Ethernet port
  - IP Address: 192.168.86.48 (DHCP from Nest)
  - Gateway: 192.168.86.1 (Google Nest)
  - Status: Connected
- **Cellular WAN (Backup)**:
  - Carrier: AT&T (was Verizon Wireless earlier)
  - Technology: 4G LTE
  - Status: Ready (SIM active but not in use)
  - RSRP: -92 dBm
  - Public IP: Available when active
- **LAN Interface**:
  - IP Range: 192.168.9.0/24
  - Gateway IP: 192.168.9.1
  - DHCP: Enabled

Test Methodology

Test Setup

1. **Initial State**: FX3110 operating on Ethernet WAN with successful internet connectivity
2. **Blocking Method**: Google Nest WiFi device-level internet blocking (simulates upstream ISP failure)
3. **Monitoring**: Real-time monitoring via custom Python script polling every 5 seconds
4. **Test Duration**: ~10 minutes of continuous monitoring while blocked

Test Tools

- **Monitoring Script**: FX3110_Monitor.py (custom Python application)
- **Dashboard**: FastAPI web dashboard showing real-time status
- **Command-line Tools**: ping, curl, ip route, ARP table inspection

---

Detailed Test Results

Test 1: Ping Connectivity Test

Command: `ping -I eth0 -c 3 8.8.8.8`

```
PING 8.8.8.8 (8.8.8.8) from 192.168.9.14 eth0: 56(84) bytes of data.
From 192.168.86.1 icmp_seq=1 Destination Host Unreachable
From 192.168.86.1 icmp_seq=2 Destination Host Unreachable
From 192.168.86.1 icmp_seq=3 Destination Host Unreachable

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2003ms
```

Result: FAILED - Internet unreachable through FX3110 Ethernet WAN **Note**: ICMP errors returned from 192.168.86.1 (Google Nest), confirming blocking at upstream router

---

Test 2: FX3110 Gateway Reachability

Command: `ping -I eth0 -c 3 192.168.9.1`

```
PING 192.168.9.1 (192.168.9.1) from 192.168.9.14 eth0: 56(84) bytes of data.
64 bytes from 192.168.9.1: icmp_seq=1 ttl=64 time=0.923 ms
64 bytes from 192.168.9.1: icmp_seq=2 ttl=64 time=0.819 ms
64 bytes from 192.168.9.1: icmp_seq=3 ttl=64 time=0.774 ms

--- 192.168.9.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2015ms
rtt min/avg/max/mdev = 0.774/0.838/0.923/0.062 ms
```

Result: PASSED - FX3110 LAN gateway is reachable **Conclusion**: FX3110 device itself is functioning normally

---

Test 3: Routing Table Analysis

Command: `ip route get 8.8.8.8`

```
8.8.8.8 via 192.168.9.1 dev eth0 src 192.168.9.14 uid 1001
    cache
```

Result: Routing is correct - Traffic directed to FX3110 (192.168.9.1) via eth0 **Conclusion**: End device routing is properly configured

---

Test 4: ARP Table Inspection

Command: `ip neigh show dev eth0`

```
8.8.8.8 FAILED
192.168.9.1 lladdr 7a:b2:8b:0c:db:85 REACHABLE
```

Result: - FX3110 gateway MAC address resolved successfully - 8.8.8.8 shows FAILED state (no ARP response, as expected for internet IP)

Conclusion: Layer 2 connectivity to FX3110 is working properly

---

Test 5: FX3110 Web Interface Status Check

Command: `curl --interface eth0 -s http://192.168.9.1/ | grep -E 'id="(internetStatus|technology|networkName)"'`

```
<div class="col textCol" id="networkName">AT&T</div>
<div class="col textCol" id="technology">Ethernet</div>
<div class="col textCol" id="internetStatus">Connected</div>
```

Result: CRITICAL FINDING **FX3110 Status**: Reports “Connected” on Ethernet WAN **Technology**: Shows “Ethernet” (not cellular)

Conclusion: The FX3110 incorrectly reports its WAN status as “Connected” despite having no internet connectivity.

Test 6: FX3110 WAN IP Configuration

Command: `curl --interface eth0 -s http://192.168.9.1/ | grep 'internetStatusIPAddress'`

```
<div class="col textCol" id="internetStatusIPAddress">192.168.86.48</div>
```

Result: FX3110 has valid DHCP lease from upstream router (192.168.86.48)

Conclusion: DHCP and Layer 3 addressing are functional

Test 7: Monitoring Log Analysis

Sample Log Entries During Blocking (TSV format):

```
Timestamp                 Success  Latency  WanStatus   WanSource  Tech      Carrier
2026-01-12 10:15:18.784   False            Connected   Ethernet   Ethernet  AT&T
2026-01-12 10:15:30.233   False            Connected   Ethernet   Ethernet  AT&T
2026-01-12 10:15:41.567   False            Connected   Ethernet   Ethernet  AT&T
2026-01-12 10:16:05.340   False            Connected   Ethernet   Ethernet  AT&T
2026-01-12 10:17:12.954   False            Connected   Ethernet   Ethernet  AT&T
```

Observations: - Ping Success: False (100% failure rate) - Latency: Empty/null (no successful pings) - WanStatus: “Connected” (FX3110 thinks it’s connected) - WanSource: “Ethernet” (no failover to cellular) - Duration: 10+ minutes with no automatic failover

Result: CRITICAL - No failover occurred despite sustained internet connectivity loss

---

Test 8: Extended Failover/Failback Behavior (Nest Block)

Procedure: 1. Continued the test with Google Nest device-level blocking still enabled. 2. Observed no successful pings while the FX3110 remained “Connected” on Ethernet. 3. Physically disconnected the Ethernet cable to force a link-down condition. 4. FX3110 failed over to Cellular (expected behavior). 5. After ~1 minute on Cellular, reconnected the Ethernet cable while the Nest block was still active. 6. After a few minutes, the FX3110 switched back to Ethernet even though the internet remained blocked.

Result: FAILBACK OCCURRED WITHOUT INTERNET **Conclusion**: Failback logic appears to prefer Ethernet based on link/DHCP presence only, without verifying end-to-end internet reachability.

Root Cause Analysis

Why Failover Did Not Occur

The FX3110’s failover detection appears to only monitor:

1. **Physical Link Status** (Layer 1)
   - Ethernet cable is connected
   - Link light is active
   - Physical layer is UP
2. **DHCP Lease** (Layer 3)
   - FX3110 successfully obtained IP address 192.168.86.48
   - DHCP lease is valid and renewed
3. **Local Gateway Reachability** (Layer 3)
   - FX3110 can ping/ARP to 192.168.86.1 (Google Nest)
   - Default gateway responds to ICMP
4. **End-to-End Internet Connectivity** (Layer 3/4)
   - FX3110 does NOT appear to test actual internet reachability
   - No detection of DNS failures
   - No detection of ping failures to public servers (google.com, kajeet.com, inseego.com)

Traffic Flow Analysis

```
End Device (RPi) → FX3110 (192.168.9.1) → Google Nest (192.168.86.1) → [BLOCKED] → Internet

   Sends packet      Receives & routes        Blocks packet         Never reaches
                    to upstream gateway                             destination
```

The Problem: - Traffic is successfully routed TO the Google Nest router - Google Nest accepts the packets (no ICMP redirect/unreachable from 192.168.86.1) - Google Nest silently drops packets destined for internet - FX3110 never detects the failure because it only checks local connectivity

---

Expected vs Actual Behavior

Expected Behavior

When an end device (POS terminal, Raspberry Pi, etc.) loses internet connectivity:

1. FX3110 should detect internet connectivity loss through active probing:
   - Periodic ping to public DNS servers (8.8.8.8, 1.1.1.1)
   - Periodic HTTP/HTTPS connectivity tests to known endpoints
   - DNS resolution tests
   - Manufacturer-specific endpoints (inseego.com, kajeet.com, google.com as mentioned)
2. After N consecutive failures (configurable threshold):
   - Mark Ethernet WAN as “Failed” or “No Internet”
   - Initiate failover to Cellular WAN
   - Update WAN status indicators
   - Log failover event
3. Periodic retry of Ethernet WAN:
   - Test Ethernet WAN connectivity every M minutes
   - Failback to Ethernet when internet connectivity restored
   - Prefer Ethernet over Cellular (cost savings)

Actual Behavior

1. FX3110 only monitors:
   - Physical link status (UP/DOWN)
   - DHCP lease status (BOUND/EXPIRED)
   - Local gateway ping (192.168.86.1 reachable)
2. FX3110 status remains:
   - WAN Status: “Connected”
   - Technology: “Ethernet”
   - No failover initiated
3. End devices:

- Complete loss of internet connectivity
- No automatic recovery via cellular backup
- Requires manual intervention or physical cable disconnect

Impact Assessment

Business Critical Scenarios

This failure mode affects critical use cases:

1. **Retail POS Systems**:
    - Cannot process credit card transactions
    - Cannot access inventory systems
    - Lost sales revenue
    - Customer service degradation
2. **Remote Monitoring Devices**:
    - Alarm systems cannot send alerts
    - Surveillance systems cannot upload footage
    - IoT sensors lose cloud connectivity
3. **Industrial Automation**:
    - SCADA systems lose connectivity
    - Remote diagnostics unavailable
    - Production downtime
4. **Emergency Services**:
    - Backup communication systems fail
    - Critical alerts not transmitted

Risk Level: CRITICAL

The FX3110 is marketed as a failover/backup solution for enterprise use cases. The inability to detect and respond to upstream internet failures defeats the primary value proposition of the device.

Test Environment Details

Hardware

- **Router**: Inseego FX3110 5G Indoor Router
- **Firmware Version**: [To be determined - need to check device]
- **SIM Card**: Kajeet IoT SIM (ICCID: 89148000012424836001)
- **Test Device**: Raspberry Pi 4B
    - OS: Linux 6.6.99
    - Ethernet: Connected to FX3110 LAN
    - WiFi: Connected to Google Nest (for SSH access)

Software

- **Monitoring Application**: FX3110_Monitor.py v2.0.0
  - Language: Python 3.11
  - Polling Interval: 5 seconds
  - Logging Format: TSV
  - Dashboard: FastAPI web interface
- **Upstream Router**: Google Nest WiFi
  - Model: [Nest WiFi Router]
  - Blocking Method: Device-level internet access control

Appendix A: Network Configuration

Raspberry Pi Network Interfaces

```
# eth0 - Connected to FX3110 LAN
inet 192.168.9.14/24
gateway 192.168.9.1

# wlan0 - Connected to Google Nest WiFi
inet 192.168.86.38/24
gateway 192.168.86.1
```

FX3110 Configuration

```
WAN Interface: Ethernet
WAN IP: 192.168.86.48
WAN Gateway: 192.168.86.1
WAN Status: Connected

LAN Interface: Ethernet
LAN IP: 192.168.9.1/24
DHCP Server: Enabled
DHCP Range: 192.168.9.2 - 192.168.9.254

Cellular Interface: Ready (standby)
Carrier: AT&T
Technology: 4G LTE
RSRP: -92 dBm
```

Appendix B: Monitoring Script Configuration

```
# FX3110_Monitor.py Configuration
BIND_INTERFACE = "eth0"          # Force traffic through FX3110
```

```
DEST = "8.8.8.8"                  # Ping target (Google DNS)
DEVICE_BASE = "http://192.168.9.1"  # FX3110 web interface
MAIN_LOOP_INTERVAL = 5            # Poll every 5 seconds
```

Appendix C: Complete Test Log Sample

```
Timestamp                   SourceIP      ActiveInterface  DestIP   Success  Latency_ms  Pub
2026-01-12 10:15:18.784    192.168.9.14  unknown          8.8.8.8  False                151.
2026-01-12 10:15:30.233    192.168.9.14  unknown          8.8.8.8  False                151.
2026-01-12 10:15:41.567    192.168.9.14  unknown          8.8.8.8  False                151.
2026-01-12 10:15:49.539    192.168.9.14  unknown          8.8.8.8  False                151.
2026-01-12 10:15:57.772    192.168.9.14  unknown          8.8.8.8  False                151.
```

Revision History

Version	Date	Author	Changes
1.0	2026-01-12	Geoffrey Noakes	Initial test report