LET’S
BUILD
TOMORROW
TODAY
How quickly do you need to detect a failure?
Data Center Network Failure Detection

Arkadiy Shapiro
Manager, Technical Marketing
Nexus 2000 – 7000

arshapir@cisco.com
@ArkadiyShapiro
Session Goals

At the end of the session, the participants should understand:

• Where failure detection fits in achieving network fast convergence

• Design aspects of network failure detection in a data center environment

• Which failure detection technologies are needed to achieve business needs and SLAs

• Advances in network failure detection technologies
Session Non-goals

This session does not include:

• Discussion on other aspects of fast convergence beyond failure detection

• Discussion on user-driven failure detection methods (ping, traceroute etc) and using scripts / EEM to automate reaction based on result / Syslog / SNMP trap

• Troubleshooting

• Detailed roadmap discussion for related Cisco products
Agenda

- Overview
- Layer 1 Failure Detection
- Layer 2 Failure Detection
- Layer 3 Failure Detection
- Additional Failure Detection Mechanisms
- Summary
Agenda

- Overview
  - Layer 1 Failure Detection
  - Layer 2 Failure Detection
  - Layer 3 Failure Detection
  - Additional Failure Detection Mechanisms
  - Summary
Routing Convergence in Action

A quick reminder…

D: I don’t care, nothing changes for me

A: Ok, fine, will use path via D

B: my link to C is down

B: Oops.. Problem

C: my link to B is down

Loss of Connectivity = \( t_4 - t_0 \)
Routing Convergence Components

1. Failure Detection
2. Failure Propagation (flooding, etc.)
3. Topology/Routing Recalculation
4. Update of the routing and forwarding table (RIB & FIB)

IGP and BGP Reaction

\[ t_0 \quad t_1 \quad t_2 \quad t_3 \quad t_4 \]
Failure Detection Overview

- Detecting the failure is **most critical** and **most challenging** part of network convergence.

- Failure Detection can occur on different levels / layers:
  - Physical Layer (1)
  - Data link Layer (2)
  - Network Layer (3)
  - Service / Application (not covered)

- Do you really need to touch all the layers?
Failure Detection Tools
Layered Approach

- **Application / Service**
  - IP SLA
  - FabricPath OAM

- **Layer 3**
  - Aggressive Timers for Various Protocols
  - BFD for BGP, OSPF, IS-IS, EIGRP, FHRPs, static and FabricPath / TRILL

- **MPLS**
  - BFD for MPLS LSPs / TE-FRR

- **Layer 2**
  - UDLD
  - LACP
  - 802.1ag CFM / Y.1731 FM
  - 802.3ah Link OAM

- **Layer 1**
  - Bit transmission
  - Signaling: Auto-negotiation / Remote Fault Indication
  - Other: Carrier Delay / Debounce
Interconnection Options

A. Layer 3 p2p
B. Layer 3 with a Layer 1 (DWDM) “bump” in wire
C. Layer 3 with a Layer 2 (Ethernet / Frame Relay / ATM switch) “bump” in wire
D. Layer 3 with a Layer 3 (Firewall / router) “bump” in wire
## Data Center Requirements

Impact on network failure detection

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Meaning for Failure Detection</th>
</tr>
</thead>
</table>
| Fast (often sub-second) network convergence       |  Sub-second detection  
 Link / path isolation                           |
| Port density and multi-tenant scale               |  High number of protocol sessions  
 Protocol offload                                  |
| High Availability                                 |  SSO / ISSU / Graceful insertion and removal                      |
| Link aggregation or wide ECMP                    |  Ability to operate on all interface types                        |
| Simple to configure and maintain                  |  More failure scenarios covered by one technology                 |

• Will traditional enterprise networking approaches apply?

• Will routing failure detection technologies apply to switching environment?
Data Center Reference Topology – VPC

Core / Edge

Aggregation / DCI / Services

Access

L2 link
FEX link
L3 link
Data Center Reference Topology – FabricPath

- FP link
- CE link
- FEX link
- L3 link
- Dark Fiber
- OTV
Data Center Reference Topology – Standalone Fabric

Details: BRKDCT-3378 - Building Data Center Networks with Overlays (VXLAN/EVPN & FabricPath)
Focus on Specific Data Center Scenarios

- **Layer 2 Classical Ethernet**
  - Single p2p link
  - Bundle

- **FabricPath**
  - Single p2p link
  - Bundle

- **Layer 3**
  - Single p2p link
  - Bundle
  - Sub-interfaces
  - SVI on top of Classical Ethernet
  - SVI on top of FabricPath
Engineering Complexity vs. Gain

K.I.S.S

Range of viable engineering options may vary by type of application

Number of possible approaches, or combinations of approaches.

Cost and Complexity

Potential Over-Engineering

Viable-Engineering

Re-engineering Required

Loss

(Impairments/Time)
Agenda

- Overview
- **Layer 1 Failure Detection**
- Layer 2 Failure Detection
- Layer 3 Failure Detection
- Additional Failure Detection Mechanisms
- Summary

Bit transmission
Signaling: Auto-negotiation / Remote Fault Indication
Other: Carrier Delay / Debounce
Layer 1 Failure Detection – Ethernet

Link Fault Signaling

- Ethernet mechanisms like auto-negotiation (1 GigE) and link fault signalling (10 GigE 802.3ae / 40 and 100 GigE 802.3ba) can signal local failures to the remote end

- Challenge to get this signal across an optical cloud as relaying the fault information to the other end is not always possible
Carrier Delay

- Running timer in software
- Standard **routing** platform feature
- Filters link **up** and **down** events, notifies protocols
- This behaviour is not desirable for Fast Convergence

```
interface ...
carrier-delay msec 0
```

- NX-OS only supports on SVI
  - Sets timer at 100 msec to suppress short flaps
  - Not recommended to set carrier-delay to 0 on SVI
Debounce Timer

- Delay link **down** notification only
- Runs in firmware
- 100 msec default in NX-OS
- Most cases recommended to keep it at default
- Standard **switching** platform feature

```bash
switch(config)interface ...
switch(config-if)# link debounce time ?
<0-5000> Timer value (in milliseconds)
```
## Carrier Delay vs Debounce timer

### General Guidance

<table>
<thead>
<tr>
<th>Carrier Delay</th>
<th>Debounce timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runs in software</td>
<td>Runs in firmware</td>
</tr>
</tbody>
</table>

**Applicable to:**
- **Routers** except Ethernet LAN switching interfaces (i.e. Cisco 7600 with WS-X6708 card)
- WAN interfaces on switches (i.e. ES+ or SIP/SPA on Catalyst 6500)
- SVIs on switches

**Filters link down and up events**

**Applicable to:**
- **Switches** except WAN interfaces (i.e. ES+ or SIP/SPA on Catalyst 6500)
- Ethernet LAN switching interfaces on routers (i.e. Cisco 7600 with WS-X6708 card)

**Filters link down events only**

---

Make sure to test before implementing!
IP Event Dampening

Isolating unstable links

Actual interface state

Accumulated penalty

Interface state seen by routing protocols

Maximum penalty

Suppress threshold

Reuse threshold

Accumulated penalty

Actual interface state

Interface state seen by routing protocols
Agenda

- Overview
- Layer 1 Failure Detection
- **Layer 2 Failure Detection**
- Layer 3 Failure Detection
- Additional Failure Detection Mechanisms
- Summary
Unidirectional Link Detection (UDLD)

- Light-weight Layer 2 failure detection protocol
- Designed for detecting:
  - **One-way** connections due to physical or soft failure
  - **Mis-wiring** detection (loopback or triangle)
- Cisco proprietary, but listed in informational RFC 5171
- Runs on any single Ethernet link, even inside bundle
- Centralized implementation in switching platforms
- **Message interval**: 7 - 90 sec (default: 15 seconds)
- **Detection**: $2.5 \times \text{interval} + \text{timeout value} (4\text{ sec}) \rightarrow \sim 41\text{ sec}$
UDLD Basics of Operation

Message Types

- **PROBE / HELLO** messages are sent periodically in steady state
  - A “resync” flag is set on the first message, triggering detection

- **ECHO** messages are sent during the Detection phase

- **FLUSH** message is sent when UDLD is disabled

- All messages identify the sender’s device and port

- **ECHO/PROBE** messages also carry additional data
  - Operating parameters of the protocol on that port
  - Device and port identifiers for any neighbor devices
UDLD Basics of Operation

Peer Discovery and Relationship

- With ECHO messages, each device learns:
  - What its connected to and peer’s message interval
  - What its neighbors think they are connected to!
- This information can then be used to detect faults
- **FLUSH** message is sent when UDLD is disabled
- Aging mechanism with PROBE messages
  - Information from neighbors that is not periodically refreshed is eventually timed out
  - This can also be used for fault detection
## UDLD Failure Reaction

### Normal vs. Aggressive mode

<table>
<thead>
<tr>
<th>Normal</th>
<th>Aggressive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set port to err-disable state in case of uni-direction condition: Empty Echo packet, Uni-direction, TX/RX loop, and Neighbor Mismatch</td>
<td>Set port to err-disable state in case of uni-direction condition: Empty Echo packet, Uni-direction, TX/RX loop, and Neighbor Mismatch</td>
</tr>
<tr>
<td>Does NOT err-disable the port in case of sudden cessation of udld packets</td>
<td>Set port to err-disable state in case of sudden cessation of UDLD packets: port is put in err-disable mode if no udld packets are received for 3 x hello-time + 5 sec (=50 secs, default)</td>
</tr>
</tbody>
</table>
UDLD Scenario 1

Empty-Echo condition or age out

- Echo Packet from A to B has “My Switch-ID A, My Port-ID e x/y”
- When B sends the echo-reply back, it is expected to have “My Switch-ID B, My Port-ID e w/z” AND “Your Switch-ID A, Your Port-ID e x/y”.
- Transmit path failure from A to B
- When B sends the echo-reply back, the echo-reply packet has only “My Switch-ID B, My Port-ID e w/z. B timed out!”
UDLD Scenario 2

Miswiring Detection

- Caused by packet flowing only in one direction
- Key differentiating factor of UDLD
- With LC fiber connection, this error is less common
UDLD Scenario

Loopback Faults

- **Condition:**
  - Rx and tx sides of a port are connected to each other
  - Packets sent on a port are received back on that port

- **Detected by:**
  - Sender’s device and port id match the local device and port id

- **Covers:**
  - Loopback of the physical link between tx and rx
  - Loopback via an intermediate (non-UDLD) device
UDLD Scenario

Loopback Faults

• TX-RX Loop:

This is a less common error type, the cause is the TX/RX of the port are mis-wired to each other, check the wire to the disabled port.
Spanning Tree Bridge Assurance

Almost like a routing protocol…

- Turns STP into a bidirectional protocol
- Ensures spanning tree fails “closed” rather than “open”
- All ports with “network” port type send BPDUs regardless of state
- If network port stops receiving BPDUs, port is placed in **BA-Inconsistent** state (blocked)

```
%STP-2-BRIDGE_ASSURANCE_BLOCK: Bridge Assurance blocking port Ethernet2/48 VLAN0700. -
switch# sh spanning vl 700 | in -i bkn
Eth2/48      Desg  BKN*4  128.304  Network P2p  *BA_Inc
```

- **NX-OS** / Nexus caveats:
  - Not recommended on VPC member ports
  - **ISSU on Nexus fixed platforms not supported with STP BA** (VPC peer-link is exception)
Without Bridge Assurance

Loop!
With Bridge Assurance

- Stopped receiving BPDUS!

%STP-2-BRIDGE_ASSURANCE_BLOCK: Bridge Assurance blocking port Ethernet2/48 VLAN0700
switch# show spanning vl 700 | in -i bkn
Eth2/48    Altn  BKN*4  128.304  Network P2p  *BA_Inc
Link Aggregation Control Protocol (LACP)
IEEE 802.1AX (formerly 802.3ad)

- Configuration consistency
- Wiring consistency

- Bundle member keepalives (LACP BPDUs)
- Detect unidirectional links

- Loss of heartbeat typically triggers port suspend

  ➢ **Note:** need to adjust default setting on Nexus 5x00/6000 interface to:
  
  `lacp suspend-individual`
LACP Slow and Fast Hellos

- Traditional LACP heartbeat intervals:
  - Long (default): 30 sec (90 sec failure detection)
  - Short (optional): 1 sec (3 sec failure detection)

- Heartbeats typically sent from supervisor, so no SSO / ISSU guarantee with aggressive timers

- **Best practice**: use long interval and configure on both peers
  - LACP does not enforce same values on both ends!

```
interface Ethernet1/7
lacp rate fast
```
UDLD “Original” Enterprise Deployment Scenarios
Still relevant to Data Center in 2015?

Figure 1: Spanning Tree Loop Prevention

Figure 2: Spanning Tree Fast Convergence

Figure 3: Ether-channel Convergence
UDLD Perspectives and Best Practices

- How much do you really need UDLD?
  - Physical uni-directional failures are communicated by Layer 1 mechanisms
  - STP Bridge Assurance to account for soft failures in either direction
  - LACP to account for failures on bundle members
  - Chance of miswiring is small
  - Layer 3 / FabricPath p2p link with bidirectional protocol already running?

- If UDLD is still needed:
  - Use normal mode and default timers
OAM

Current Protocol Positioning

- Link OAM - Any point-to-point 802.3 link
- CFM / Y.1731 - End-to-End UNI to UNI
- E-LMI - User to Network Interface (UNI)
- MPLS OAM - within MPLS cloud
Ethernet OAM

Building Blocks

- **IEEE 802.3ah (clause 57)**
  - Ethernet Link OAM
  - Also referred as 802.3 OAM or **Link OAM**

- **IEEE 802.1ag**
  - Connectivity Fault Management (CFM)
  - Also referred as Service OAM

- **ITU-T Y.1731**
  - OAM functions and mechanisms for Ethernet-based networks

- **MEF E-LMI**
  - Ethernet Local Management Interface
Ethernet Link OAM

IEEE 802.3ah (IEEE 802.3-2008, Clause 57)

- Provides mechanisms for “monitoring link operation”
- Runs on any single point-to-point Ethernet link
- Uses “Slow Protocol” frames called OAMPDUs
  - OAMPDU interval: 100 msec – 1 sec (1-10 pps)
  - Minimum Timeout: 200 msec (IOS XR), 2 sec (IOS)
- Extensible and flexible protocol
- Current support mainly on following platforms: Cisco 7600, ASR 9000, ASR 901, ASR 903, ME switches

(1) No more than 10 frames transmitted in any one-second period
Ethernet Link OAM

Key Functions for Future Data Center

- **OAM Discovery**
  - Discover OAM support, peer identity and capabilities per device
  - Timeout

- **Link Monitoring**
  - Basic error definitions for Ethernet so entities can detect degraded links and isolate them

- **Remote Failure Indication**
  - Mechanisms for one entity to signal another that it has detected an error

- **Remote Loopback**
  - Used to troubleshoot networks, allows one station to put the other station into a state whereby all inbound traffic is immediately reflected back onto the link

- **Remote MIB Variable Retrieval**
  - Ability to read one or more MIB variables from the remote DTE
Link OAM PDU Format

• All slow protocols use Ethertype 88-09
• link-layer OAM is differentiated by a sub-type of 03
• OAMPDUs are multicast to a specific multicast address that is link-constrained, since OAMPDUs only traverse a single link, but are never forwarded by bridges or switches
• OAMPDUs contain control and status information needed to monitor, test and troubleshoot OAM-enabled links. Encoded in TLV format
Link OAM Discovery

- First phase of Ethernet OAM
- Discovery has a simple state machine:
  - Send Information OAMPDU in a periodic fashion
  - Discover peer device and its OAM configuration and capabilities
  - Decide whether OAM clients can be fully operational on the link
  - Detect timeout based on lack of OAMPDUs from peer
  - No message interval exchange or negotiation!

```
switch#show ethernet oam discovery
interface fas 1/1
FastEthernet1/1
Local client
Administrative configurations:
  Mode:              active
  Unidirection:      not supported
  Link monitor:      supported (on)
  Remote loopback:   not supported
  MIB retrieval:     not supported
  Mtu size:          1500

Operational status:
  Port status:       operational
  Loopback status:   no loopback
  PDU revision:      0

Remote client
  MAC address: 0011.9321.1640
  Vendor(oui): 00000C(cisco)

Administrative configurations:
  PDU revision: 1
  Mode: active
  Unidirection: not supported
  Link monitor: supported
  Remote loopback: not supported
  MIB retrieval: not supported
  Mtu size: 1500
```
### Link OAM

#### Additional Troubleshooting

```
show ethernet oam summary
show ethernet oam discovery [interface <interface name>]
```

```
switch#show ethernet oam summary
Symbols:
* - Master Loopback State, # - Slave Loopback State
Capability codes: L - Link Monitor, R - Remote Loopback
U - Unidirection, V - Variable Retrieval

<table>
<thead>
<tr>
<th>Local Interface</th>
<th>Remote MAC Address</th>
<th>Remote OUI</th>
<th>Mode</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fa1/1</td>
<td>0011.9321.1640</td>
<td>00000C</td>
<td>active</td>
<td>L</td>
</tr>
<tr>
<td>Fa1/2</td>
<td>0011.9321.1640</td>
<td>00000C</td>
<td>active</td>
<td>L</td>
</tr>
<tr>
<td>Fa1/3</td>
<td>0011.9321.1640</td>
<td>00000C</td>
<td>active</td>
<td>L R</td>
</tr>
<tr>
<td>Fa1/4</td>
<td>0011.9321.1640</td>
<td>00000C</td>
<td>passive</td>
<td>L</td>
</tr>
</tbody>
</table>
```

```
switch#show ethernet oam status int fas 4/4
FastEthernet4/4
General
Admin state: enabled
Mode: passive
PDU max rate: 5 packets per second
PDU min rate: 1 packet per 2 seconds
Link timeout: 30 seconds
High threshold action: no action
Link fault action: error disable
interface
Dying gaps action: error disable
interface
Critical event action: error disable
interface

Link Monitoring
Status: supported (on)

Symbol Period Error
Window: 100 x 1000000 symbols
Low threshold: 1 error symbol(s)
High threshold: none

Frame Error
Window: 10 x 100 milliseconds
Low threshold: 1 error frame(s)
High threshold: none
```

---

Cisco Live!
Link OAM scale and ISSU

- **Scale**
  - Slow protocol but 100 msec interval for all ports on a linecard is not slow!
  - Protocol offload to I/O module CPU helps
  - Protocol offload to FPGA (ME 3400) helps even more!

- **ISSU (the “zero service disruption one”😊)**
  - Need graceful protocol mechanisms to support SSO / ISSU – standard does not specify
  - Not possible to inflate timers since timers are not negotiated!
Link OAM Basic Configuration

IOS and IOS XR

interface TenGigE 0/1/0/0
  ethernet oam
  hello-interval 100ms
  connection timeout 2

interface TenGigEthernet4/1
  ethernet oam
  max-rate 10
  ethernet oam
  timeout 2
Link OAM Miswiring Detection (IOS XR only)

Closing the gap with UDLD

- Mechanism to detect miswiring of Ethernet ports
- Similar to UDLD, but using standard protocol with Cisco vendor extension
- Uses existing 4-byte field in periodic OAMPDU (Information OAMPDU → Information TLV → ‘Vendor Information’ field)
- Vendor Information is copied back by the peer, allowing for MWD
- Interoperates with other 802.3ah-compliant vendors
Link OAM - Link Monitoring

- Monitor link quality every 1 sec (min interval)
- Conditions monitored:
  - Errored Symbol Period
  - Errored Frame
  - Errored Frame Period
  - Errored Frame Seconds
  - Receive CRC (Cisco defined – IOS only)
  - Transmit CRC (Cisco defined – IOS only)
- Configure error condition thresholds to:
  - Signal peer with “Event Notification” OAMPDU
  - Syslog / SNMP trap
  - Isolate the link
Link OAM – Link Monitoring

Example: Frame Error Detection and Link Isolation on IOS

- **Problem**
  - Ensure frame errors don’t propagate through the network and degrade application performance
  - Need to operate with or without neighbor discovery

- **Solution**
  - IEEE 802.3ah for link monitoring and error-disable

```plaintext
interface GigabitEthernet1/1
  ethernet oam
  ethernet oam link-monitor frame window 1000
  ethernet oam link-monitor frame threshold high 10
  ethernet oam link-monitor high-threshold action error-disable-interface

Nov 10 09:56:08.643: EOAM LM(Gi1/1): sending an EventTLV!
Nov 10 09:56:09.643: %ETHERNET_OAM-5-LINK_MONITOR: 94 rx frame errors detected over the last 1 seconds on interface Gi1/1.
Nov 10 09:56:09.643: EOAM LM(Gi1/1): sending an EventTLV!
Nov 10 09:56:09.647: %PM-SP-4-ERR_DISABLE: link-monitor-failure error detected on Gi1/1, putting Gi1/1 in err-disable state
```
Link OAM – Failure Reaction

Path Isolation

- **No standards that define this!**
- Depending on implementation, available options for failure reaction / path isolation:
  - Syslog / SNMP trap
  - Signal peer using specific OAMPDU
  - **Error-disable**
  - **Error-block** / Ethernet Failure Detection (EFD)
- **Error-disable** – operate at Layer 1, useful when need to force manual intervention after error (like miswiring)
Link OAM Failure Reaction

Path Isolation with Ethernet Failure Detection (EFD)

- Allows OAM to bring interface “line protocol” down when a problem is detected
- Interface is “down” to higher protocols (STP, IGPs, BGP) – will trigger reconvergence
- OAM protocols continue to operate
- Automatic and deterministic recovery when fault is resolved
- Reduced interface up/down churn
- IOS XR only, can not be used with link monitoring
Ethernet Failure Detection (EFD)

Logical Diagram

- L2VPN
- IPv4
- IPv6
- MPLS

- MAC layer
- EFD
- Link OAM
- CDM
- Packet I/O

- Interface

SW Support
IOS XR
• 3.9
Link OAM

Ready for Data Center?

- Link OAM could be adopted in future Data Center

- Rely on STP BA / LACP / UDLD for now:
  - Link OAM **path isolation** based on **timeout** only on IOS XR
  - Link OAM **miswiring detection** only on IOS XR as proprietary extension

- Consider Link OAM today for Data Center edge if:
  - Existence of supported hardware
  - Requirement to adhere to standard protocols
  - **Link Monitoring capabilities**

Details: BRKNMS-2202 - Ethernet OAM: Technical Overview and Deployment Scenarios
Agenda

- Overview
- Layer 1 Failure Detection
- Layer 2 Failure Detection
- Layer 3 Failure Detection
- Additional Failure Detection Mechanisms
- Summary
Failure Detection at Layer 3

- In some cases, failure detection relies on checks at Layer 3
- How quickly can I detect a failure (neighbor down event)?
Is Layer 3 Failure Detection Tuning Necessary?

- Needed when:
  - Intermediate L2 hop over L3 link
  - Concerns over any protocol software failures
  - Concerns over unidirectional failures (link or ASIC) on point-to-point physical L3 links

- May not be needed when:
  - Point-to-point physical L3 links with no concerns over unidirectional failures
  - Enough software redundancy to account for protocol software failures
  - FHRPs are running in active-active mode (VPC / VPC+ / Anycast HSRP)
Active/Active Forwarding Mode for FHRPs in NX-OS

- HSRP, VRRP and GBLP in vPC / vPC+ environment operate in Active/Active mode
- Anycast HSRP operates in 4-way active mode in FabricPath environment
- No additional configuration required
- General best practices still apply, except:
  - Since running in active/active mode, no BFD or aggressive timers are needed
  - No need to manipulate priorities / preemption on different devices to achieve load-balancing
Layer 3 Failure Detection

Protocol Timers

- All Layer 3 protocols (FHRPs, BGP, IGPs etc) exchange periodic hellos to:
  - Maintain adjacencies (pass protocol specific info)
  - **Check neighbour reachability and detect failure**

- Hello/Keepalive and Dead/Hold timers can be tuned down, however it is not recommended:
  - Each interface may have 2-3+ protocols establishing adjacency (e.g. HSRP, PIM, OSPF on SVI)
  - Configuration complexity and waste of link bandwidth
  - Increased supervisor CPU utilization → false-positives
  - Challenges supporting ISSU / SSO
  - Challenges achieving sub-second detection
  - **Having said this:** works reasonably well in small & controlled environments
Bidirectional Forwarding Detection (BFD)

RFC 5880 / 5881

- **Lightweight** hello protocol designed to run over multiple transport protocols:
  - IPv4, IPv6, MPLS, FP / TRILL etc.

- Designed for sub-second failure detection

- Any interested client (OSPF, BGP, HSRP etc.) registers with BFD and is notified as soon as BFD detects a neighbor loss

- All registered clients benefit from **uniform failure detection**

- Runs on physical, virtual and bundle interfaces

- UDP port 3784 / 3785 (for echo)
Layer 3 Failure Detection with BFD

- **Bidirectional Forwarding Detection (BFD)** – strongly recommended over aggressive protocol timers

- **General advantages:**
  - Reduced control plane load and link bandwidth usage
  - Sub-second failure detection
  - In-flight timer negotiation

- **NX-OS and Nexus advantages:**
  - Stateful restart, SSO and **ISSU** support
  - Protocol off-load / distributed implementation – I/O module transmits / receives BFD packets
  - Per-link implementations with bundles
BFD Peer Establishment

Timer Negotiation

• No discovery – peer IP provided by client!
• Neighbors continuously negotiate their desired transmit and receive rates in terms of microseconds.
• The system reporting the slower rate determines the transmission rate.

**Negotiate rates**

Desired Receive rate = 50 ms
Desired Transmit rate = 100 ms

Desired Receive rate = 60 ms
Desired Transmit rate = 40 ms

Green Transmits at 100ms
Orange transmits at 50ms

```
interface <name>
  bfd interval <msec> min_rx <msec> multiplier <n>
```
BFD Operation Modes

- Session established using asynchronous control packets

- Asynchronous mode (no echo):
  - Control packets sent at negotiated rate
  - Independent session
  - Neighbor declared dead if no packet is received for <interval * multiplier> period

- Additionally, if echo is negotiated:
  - Control packets sent at slow rate
  - Self-directed echo packets sent at negotiated rate (max (Desired Tx, echo Rx)), used for failure detection
### BFD – OSPF Interaction

**Example**

- **X-** Forwarding plane failure between R1 and R2
- **X-** BFD detects failure between R1 and R2
- **X-** OSPF adjacency reset between R1 and R2
BFD in NX-OS and Nexus switches

- Version 1
- Single hop
- IPv4 and IPv6 (except Nexus 5x00/6000), single-hop
- Packets sent with DSCP CS6 / CoS 6
- Nexus 3000/3100/3500/5500/5600/6000/9300 with centralized implementation
- Nexus 7x00/9500 with distributed implementation
- Nexus 2000 ports (L2 or L3) do not support BFD

NX-OS Software Support

<table>
<thead>
<tr>
<th>Nexus 3000</th>
<th>5.0(3)U2(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nexus 3500</td>
<td>6.0(2)A4(1)</td>
</tr>
<tr>
<td>Nexus 5500/6000</td>
<td>6.0(2)N2(1)</td>
</tr>
<tr>
<td>Nexus 5600</td>
<td>7.0(1)N1(1)</td>
</tr>
<tr>
<td>Nexus 7000</td>
<td>5.0(2a)</td>
</tr>
<tr>
<td>Nexus 7700</td>
<td>6.2(2)</td>
</tr>
<tr>
<td>Nexus 9000</td>
<td>6.1(2)I1(1)</td>
</tr>
</tbody>
</table>
BFD Off-load

Addressing higher session scale

- **SUP-BFD** - BFD process running on Supervisor Engine
  - Interfaces with LC-BFD processes
  - Interfaces with BFD clients

- **LC-BFD** – BFD process running on CPU of each I/O module
  - Communicates with SUP-BFD process
  - Generates BFD hellos (echo and async)
  - Receives BFD hellos from peer (async)

- Nexus 7x00 additional features
  - Support for stateful restart, SSO and ISSU
  - Offload to hardware accelerator on F3 I/O module*

* - From NX-OS 7.2

---

Check [Nexus 7000](#) and [Nexus 9000](#) Verified Scalability Guides for latest scalability information.
BFD Sub-interface Optimization

Scaling up number of sessions per module

• Single session selected among all v4 / v6 sessions to run at fastest interval
• Rest of sessions run at slow interval (default: 2 sec)
• Failure of a “fast” session signaled to all “slow” sessions
• **Use-case**: high scale multi-tenancy with more routing protocol sessions than BFD sessions supported on I/O module
Layer 3 Fast Failure Detection and Link Bundles

Challenges

- Scenarios:
  1. Layer 2 bundle between 2 SVIs
  2. Layer 3 bundle

- Hash algorithm used to distribute the load across bundle members

- Chances are high that control plane packets are only carried on a single link:
  - Can’t reliably test all links
  - Single bundle member malfunction can cause black holes which remain undetected
  - Rely on Layer 1 or Layer 2 detection

- Can use parallel Layer 3 links instead, load-sharing properties are often similar

- BFD single session VS per-link sessions?
BFD Logical Mode
Nexus 3000 / 5500 / 5600 / 6000 / 7x00 / 9x00

- **Single** BFD session per L3 destination address
- Internal algorithm to determine which I/O module hosts BFD session
- BFD packet distribution:
  - Prior to NX-OS 5.2(1) – Tx packets are polarized on one bundle link per session
  - From NX-OS 5.2(1) – Tx packets are round-robin load-balanced on all bundle links (**except N9K**)  
- Rx packets are always polarized on one bundle link per session
- Async + echo
- Minimum interval is 250 msec x 3
BFD Per-link Mode

Nexus 5600 / 6000 / 7x00 / 9x00

- BFD session per port-channel member
- Master session on SUP consolidates member states and communicates with clients
- LACP is required for port-channels
- Async only, no echo
- Layer 3 port-channel / sub-interface only
- Minimum interval: 50 msec x 3
BFD over Bundle Members (BOB)
CRS / ASR 9000 / XR 12000

- IPv4 BFD session per bundle member
- IPv6 relies on IPv4 session state
- Verify every member link forwarding state by establishing BFD session before its added to bundle
- Master session on RP consolidates member states and communicates with clients
- Async + echo
- Ethernet and POS bundles
- IOS XR proprietary

interface bundle-ether 1
  bfd
  address-family ipv4
  fast-detect
  minimum-interval 15
  multiplier 3
  destination 10.11.12.13
BFD Logical Bundles (BLB)

- Single BFD session per L3 destination address
- Internal algorithm to decide which I/O module hosts BFD session
- BFD packet distribution - **Tx and Rx packets are polarized on one bundle link per session**
- IPv4 and IPv6 sessions
- Async only
- Replaces BVLAN mode but backward compatible!
- **Verified interoperability with IOS and NX-OS single session modes**
- Minimum interval: 50 msec x 3 (depends on linecard)
BFD Interoperability with Bundles

- Use-case: Nexus – non-Nexus at DC edge
- Standards did not address until RFC 7130
- Single session
  - Easiest to achieve with current implementations
  - Verified interoperability between IOS XR BLB mode, IOS and NX-OS single session mode
- Per-link sessions
  - Recommended, but often platform proprietary
  - RFC 7130 addresses interoperability!
BFD and Nexus Virtual Port-Channel (VPC)

- BFD over VPC (MCEC) – due to no L3 over VPC support

- BFD over VPC peer-link support:
  - Nexus 7x00 M1, M2 or F3 peer-link
  - Nexus 7x00 F2/F2e peer-link in F2/F2e VDC
  - Nexus 7x00 F2e peer-link in M-F2e VDC
  - Nexus 3000 / 3100 / 9x00 (no echo)
  - Nexus 5500/5600/6000
BFD for FabricPath

FabricPath IS-IS as BFD client

- Current support on Nexus 5500 / 5600 / 6000 / 7x00 (from NX-OS 7.2)
- **Use-case:** peer switch path failure detection
- Asynchronous mode only
- Based on **RFC 7175**
  - Does not include bundle per-link, so proprietary implementation
  - BFD TLV (**RFC 6213**) - IS-IS notifies BFD of Rbridge IDs and **waits** for BFD session to come up before reaching full adjacency

```
switch(config)# fabricpath domain default
switch(config-fabricpath-isis)# bfd fabricpath interval 500 min_rx 250 multiplier 3
switch# sh bfd neighbors fabricpath int port-channel 2402
```

OurAddr     NeighAddr     LD/RD     RH/RS    Holdown(mult)   State   Int
002a.6a66.88bc  002a.6a1c.929c  90519044/1090519045 Up     1047(3)   Up     Po2402
BFD for FabricPath and TRILL

Point-to-Point vs. Shared Ethernet segment

- TRILL specifies support shared Ethernet segment with several peers
- FabricPath can only peer on point-to-point links
- BFD may be more needed for TRILL than FabricPath except…
FabricPath and Failure Detection

Back to Layer 1 “bump” in the wire…

- Multi-POD / multi-building interconnect scenarios may require BFD
BFD over FabricPath

FabricPath as Transport for BFD

- Routing protocol / FHRP peering over FabricPath network
- BFD between SVIs or sub-interfaces

* - From NX-OS 7.2
BFD over FabricPath

Example and Challenges

- HSRP on CE connected devices
- Forwarding path testing?
  - HSRP – multicast following multi-destination trees
  - BFD – unicast
  - Potentially different paths
- Recommendation to increase BFD timers
  - Account for any failures in FabricPath network and re-convergence
- So we can follow these recommendations or…evolve FabricPath to Standalone Fabric!
BFD for IP Unnumbered

Failure Detection for Standalone Fabric (VXLAN) Underlay

Multiple Fabric facing Ethernet Interface (IP Unnumbered to Loopback)

Single Loopback acting as VTEP

router ospf 1
  bfd

interface loopback0
  ip address 1.1.1.1/32

interface e1/1-4
  ip un-numbered loopback0
  ip router ospf 1 area 0

NX-OS 7.2 on Nexus 7x00
NX-OS 7.1 for Nexus 5600/6000
BFD for OTV

Peer Edge Device Failure Detection

- No more waiting for IS-IS timeout!
- Relies on hardened multi-homing
  - BFD between site VLAN SVIs for detection within the site
  - Route reachability to neighbor IP address for detection on overlay
- Both adjacencies must be down to declare ED failure
  - Site – 150 msec detection
  - Overlay – < 5 sec (depends on routing performance and tuning in overlay)

```bash
otv site-vlan 1001
  otv isis bfd
interface Vlan1001
  ip address 10.0.0.1/30
  bfd interval 50 min_rx 50 multiplier 3
```
BFD for Static Routes

Single Hop Example

- Use-case: gateway / next-hop liveliness detection
- **Fail-close solution** (remove static route and not reinstate until BFD is up)
- Must be configured on both ends

```
ip route 0.0.0.0/0 Vlan 10 20.0.0.1
ip route static bfd Vlan 10 20.0.0.1
```

```
ip route 30.0.0.0/24 Vlan 20 10.0.0.1
ip route static bfd Vlan 20 10.0.0.1
```

```
switch# sh ip route
0.0.0.0/0, ubest/mbest: 1/0
*via 20.0.0.1, Vlan 10, [1/0], static
```

```
switch# sh ip route
30.0.0.0/0, ubest/mbest: 1/0
*via 10.0.0.1, Vlan 20, [1/0], static
```
BFD Multihop

RFC 5883

- BFD peer is not directly connected on Layer 3
- Use-case 1: eBGP multihop
- Use-case 2: firewall / NAT load-balancing with static route
IP SLA tracking for Static Route or PBR

Alternative to BFD Multi-hop with more capabilities

• Establish IP SLA probe as an object
• PBR path or static route availability is dependent on IP SLA state or reachability
• PBR is done in hardware on Nexus switches
• Use-case: firewall / NAT load-sharing with PBR or static route
IP SLA tracking for PBR

Configuration Example

feature ip sla sender
feature pbr

ip sla 1
    icmp-echo 10.1.1.1
    timeout 2500
    frequency 3

track 1 ip sla 1 reachability

route-map PBR permit 10
    match ip address ACL
    set ip next-hop verify-availability 11.0.0.1 10 track 1
    set ip next-hop 12.0.0.1

interface eth1/1
    ip policy route-map PBR

ICMP-echo probe with 2.5 sec timeout
Track object for IP SLA reachability
Route-map for PBR:
1) Go to 11.0.0.1 if IP SLA is reachable
2) Go to 12.0.0.1 if IP SLA is not reachable
Apply route-map on ingress interface
BFD and Security

- Disable platform hardware security mechanisms for BFD echo to function:
  - uRPF (per interface)
    - no [ip|ipv6] verify unicast source reachable-via [any|rx]
  - IDS checks (global)
    - no hardware ip verify address identical
  - IP redirects (per interface)
    - no ip redirects
    - no ipv6 redirects

- Open rules to allow echo packets though firewall or enable loopback as source IP:
  - bfd echo-interface <a_loop_back_interface>
Layer 3 Failure Detection Best Practices
Aligning to Data Center Requirements

1. If Layer 3 fast failure detection is needed, use BFD for all protocols
2. If cant use BFD, verify aggressive timer support
3. Always plan your BFD scale and check with platform capabilities (centralized vs distributed architecture, interface and client support locally and on peer)
4. Use BFD echo (default) whenever possible, check security
5. On Layer 3 port-channels, use per-link mode and prefer that over echo
6. BFD single-hop for BGP – make sure neighbor update source is a directly connected interface
7. Make sure BFD packets are prioritized appropriately (Marked with IP precedence 6 / DSCP CS6 / CoS 6, can also be classified by udp 3784+3785)
Agenda

- Overview
- Layer 1 Failure Detection
- Layer 2 Failure Detection
- Layer 3 Failure Detection
- Additional Failure Detection Mechanisms
- Summary
Remote Integrated Service Engine (RISE)

- Logical integration of a service appliance with Nexus platforms (5500/5600/6000/7x00/9x00)
- Initial support for Citrix ADC and Cisco NAM physical and virtual appliances
- Communication protocol between Nexus switch and appliance:
  - Configuration push
  - Keepalive: 5 sec interval, <= 30 sec failure detection
Nexus 2000 Fabric Extender

Satellite Discovery Protocol

- FEX discovery by parent switch and subsequent keepalives
- Operates on every physical FEX fabric port
- Keepalive interval - 3 sec
- Failure Detection – <= 30 sec (pre-defined multiplier)
- LACP not supported if port-channel is used
Generic OnLine Diagnostics (GOLD)

- Switch internal tests to detect system failures
- Interval – several min, depending on test
- Cover areas where BFD / OAM do not reach
- From Nexus 7x00 NX-OS 6.2(8), configure corrective actions
- PortLoopback, SnakeLoopback or other tests show port failure after multiple tries
  - Syslog
  - Error-disable offending port

```
switch# config t
switch(conf-t)# diagnostic eem action conservative
```
Agenda

- Overview
- Layer 1 Failure Detection
- Layer 2 Failure Detection
- Layer 3 Failure Detection
- Additional Failure Detection Mechanisms
- Summary
Summary of Recommendations

- **Layer 2 Classical Ethernet**
  - Single p2p link
  - Bundle

- **FabricPath**
  - Single p2p link
  - Bundle

- **Layer 3**
  - Single p2p link
  - Bundle
  - Sub-interfaces
  - SVI on top of Classical Ethernet
  - SVI on top of FabricPath
Summary

What have we learned?

- Fast Failure Detection is **Key** to Fast Convergence
- Match Data Center requirements to the right set of technologies
  - One protocol may be enough – keep it simple!
  - Evolving field with industry and Cisco innovations
- Design your network to take advantage of best practices
## Related Cisco Live San Diego 2015 Events

<table>
<thead>
<tr>
<th>Session-ID</th>
<th>Session Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRKRST-3363</td>
<td>Routed <strong>Fast Convergence</strong></td>
</tr>
<tr>
<td>BRKNMS-2202</td>
<td><strong>Ethernet OAM</strong>: Technical Overview and Deployment Scenarios</td>
</tr>
<tr>
<td>BRKCRS-2042</td>
<td>Highly Available Wide Area Network Design</td>
</tr>
<tr>
<td>BRKDCT-1890</td>
<td>Network visibility using advanced Analytics in Nexus switches</td>
</tr>
</tbody>
</table>
Participate in the “My Favorite Speaker” Contest

Promote Your Favorite Speaker and You Could Be a Winner

• Promote your favorite speaker through Twitter and you could win $200 of Cisco Press products (@CiscoPress)

• Send a tweet and include:
  • Your favorite speaker’s Twitter handle @ArkadiyShapiro
  • Two hashtags: #CLUS #MyFavoriteSpeaker

• You can submit an entry for more than one of your “favorite” speakers

• Don’t forget to follow @CiscoLive and @CiscoPress

• View the official rules at http://bit.ly/CLUSwin
Complete Your Online Session Evaluation

• Give us your feedback to be entered into a Daily Survey Drawing. A daily winner will receive a $750 Amazon gift card.

• Complete your session surveys though the Cisco Live mobile app or your computer on Cisco Live Connect.

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
TOMORROW starts here.
References

BFD

• Building Highly Available Layer 3 Networks with Cisco NX-OS Software
• Bidirectional Forwarding Detection - Cisco Systems
• IETF BFD Working Group
• draft-ietf-trill-rbridge-bfd-07 - TRILL (Transparent Interconnetion of Lots of Links): Bidirectional Forwarding Detection (BFD) Support
• RFC 7130 - Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces
• Configuring Bidirectional Forwarding Detection on the Cisco ASR 9000 Series Router [Cisco ASR 9000 Series Aggregation Services Routers] - Cisco Systems
• Configuring Bidirectional Forwarding Detection on Cisco IOS XR Software [Cisco Carrier Routing System] - Cisco Systems
• Configuring Bidirectional Forwarding Detection [Cisco Nexus 7000 Series Switches] - Cisco Systems
• IP Routing BFD Configuration Guide, Cisco IOS Release 15.2S [Cisco IOS 15.2S] - Cisco Systems
References

OAM and UDLD

- Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide - Configuring Ethernet OAM on the Cisco ASR 9000 Series Router [Cisco ASR 9000 Series Aggregation Services Routers] - Cisco Systems

- IEEE 802.1: 802.1ag - Connectivity Fault Management

- Overview of Ethernet Operations, Administration, and Management [Cisco 7600 Series Routers] - Cisco Systems

- RFC 6291 - Guidelines for the Use of the "OAM" Acronym in the IETF

- Borderless Campus 1.0 Design Guide - Deploying High Availability in Campus

- Catalyst 4500 Series Switch Software Configuration Guide, 12.2(54)SG - Configuring UDLD

- Catalyst 6500 Release 12.2SX Software Configuration Guide - UniDirectional Link Detection (UDLD)

- Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.x - Configuring Basic Interface Parameters

- RFC 5171 - Cisco Systems UniDirectional Link Detection (UDLD) Protocol
Acronyms

• BER – Bit Error Rate
• BFD – Bidirectional Forwarding Detection
• CRC – Cyclic Redundancy Check
• DCI – Data Center Interconnect
• DFA – Dynamic Fabric Automation
• EFD – Ethernet Failure Detection
• FEX – Fabric EXtender
• FIB – Forwarding Information Base
• FSA – Fabric Services Accelerator
• GOLD – Generic OnLine Diagnostics
• ISSU – In-Service Software Upgrade
• ITD – Intelligent Traffic Director
• LACP – Link Aggregation Control Protocol
• LAN – Local Area Network
• OAM – Operations And Management
• PBR – Policy Based Routing
• PDU – Protocol Data Unit
• RIB – Routing Information Base
• RISE – Remote Integrated Service Engine
• RPF – Reverse Path Forwarding
• SDP – Satellite Discovery Protocol
• SSO – Stateful SwitchOver
• TLV – Type Length Value
• UDLD – Uni-directional Link Detection
• VDC – Virtual Device Context
• VPC – Virtual Port-Channel
• VPN – Virtual Private Network
• VXLAN – Virtual eXtensible LAN
BFD Clients and Interface Types

Nexus 7x00

- Supported BFD IPv4 clients:
  - OSPFv2
  - EIGRP
  - IS-IS
  - PIM
  - BGP
  - HSRP
  - VRRPv2
  - Static routes
  - MPLS TE FRR
  - OTV IS-IS

- Supported BFD IPv6 clients:
  - OSPFv3
  - EIGRP
  - IS-IS
  - PIMv6
  - BGP
  - Static

- Supported Layer 3 interface types:
  - Switched Virtual Interface (SVI)
  - Physical Port and sub-interface
  - Port Channel (per-link and logical modes) and sub-interface

Note: Check configuration guides and release notes for latest supported configuration
BFD Clients and Interface Types

Nexus 3000 / 3100

- Supported BFD IPv4 clients:
  - OSPFv2
  - BGP
  - Static

- Supported Layer 3 interface types:
  - Switched Virtual Interface (SVI)
  - Physical Port and sub-interface
  - Port Channel (logical mode only) and sub-interface

- Supported BFD IPv6 clients:
  - OSPFv3
  - BGP
  - Static

Note: Check configuration guides and release notes for latest supported configuration.
BFD Clients and Interface Types

Nexus 3500

• Supported BFD IPv4 clients:
  ➢ BGP
  ➢ PIM

• Supported Layer 3 interface types:
  ➢ Switched Virtual Interface (SVI)
  ➢ Physical Port and sub-interface
  ➢ Port Channel (logical mode only) and sub-interface

Note: Check configuration guides and release notes for latest supported configuration
BFD Clients and Interface Types

Nexus 5500 / 5600 / 6000

- Supported BFD IPv4 clients:
  - OSPF v2
  - BGP
  - EIGRP
  - PIM
  - HSRP
  - VRRP
  - Static route

- Supported Layer 3 interface types:
  - Switched Virtual Interface (SVI)
  - Physical Port and sub-interface
  - Port Channel (logical mode, per-link only for Nexus 6000) and sub-interface

Note: Check configuration guides and release notes for latest supported configuration
BFD Clients and Interface Types

Nexus 9000

• Supported BFD IPv4 clients:
  - OSPF v2
  - BGP
  - EIGRP
  - PIM
  - HSRP
  - VRRP
  - Static route

• Supported Layer 3 interface types:
  - Switched Virtual Interface (SVI)
  - Physical Port and sub-interface
  - Port Channel (logical mode and per-link) and sub-interface

• Supported BFD IPv6 clients:
  - OSPF v3
  - EIGRP
  - IS-IS
  - BGP
## Protocol Comparison

<table>
<thead>
<tr>
<th></th>
<th>BFD</th>
<th>UDLD</th>
<th>Link OAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OSI Layer</strong></td>
<td>L3</td>
<td>L2</td>
<td>L2</td>
</tr>
<tr>
<td><strong>Standard</strong></td>
<td>IETF RFC 5880 / 5881 (with some Cisco enhancements)</td>
<td>Cisco proprietary</td>
<td>IEEE 802.3-2008 (with some Cisco enhancements)</td>
</tr>
<tr>
<td><strong>Failures Detected</strong></td>
<td>Uni-directional soft failures</td>
<td>Uni-directional soft failures</td>
<td>Uni-directional soft failures</td>
</tr>
<tr>
<td></td>
<td>Bidirectional soft failures</td>
<td>Bidirectional soft failures</td>
<td>Bidirectional soft failures</td>
</tr>
<tr>
<td></td>
<td>Mis-wiring Detection</td>
<td>Error-disable (depending on mode)</td>
<td>Mis-wiring Detection (IOS XR)</td>
</tr>
<tr>
<td><strong>Failure Reaction</strong></td>
<td>Notify peer and clients</td>
<td>Error-disable (depending on error type and platform)</td>
<td>Notify peer</td>
</tr>
<tr>
<td></td>
<td>Remove link from bundle (IOS XR, IETF standard in future)</td>
<td>Error-block</td>
<td>Error-block</td>
</tr>
<tr>
<td></td>
<td>BFD dampening (IOS XR)</td>
<td>Ethernet Failure Detection (IOS XR)</td>
<td></td>
</tr>
<tr>
<td><strong>Bundles and Virtual Interfaces</strong></td>
<td>Bundle logical, bundle per-link, SVI, sub-interface</td>
<td>Single L2 links</td>
<td>Single L2 links</td>
</tr>
<tr>
<td><strong>Message Interval and Timeout</strong></td>
<td>Configurable, exchanged and negotiated</td>
<td>Configurable and exchanged</td>
<td>Configurable, not exchanged</td>
</tr>
<tr>
<td></td>
<td>Timeout generally in msec</td>
<td>Timeout generally in 20+ seconds</td>
<td>Timeout generally in 2+ seconds</td>
</tr>
<tr>
<td><strong>ISSU Support</strong></td>
<td>Yes - timer inflation</td>
<td>Yes with default timers</td>
<td>No (can be extended in future)</td>
</tr>
</tbody>
</table>
LET’S BUILD TOMORROW TODAY