Your Time Is Now
SR Traffic Engineering

Clarence Filsfils
Cisco Fellow - cf@cisco.com
BRKRST-3122

Strictly Confidential
Industry at large backs up SR

- Strong customer adoption: WEB, SP, Enterprise
- Standardization: IETF
- De-Facto SDN Architecture
- Multi-vendor Consensus: Interop testings
Stay Up-To-Date

segment-routing.net

linkedin.com/groups/8266623

twitter.com/SegmentRouting

facebook.com/SegmentRouting/
IETF key document for SR-TE

Segment Routing Policy for Traffic Engineering
draft-filsfils-spring-segment-routing-policy-00.txt
SR Traffic Engineering

SR-TE
RSVP-TE

- Little deployment and many issues
- Not scalable
  - Core states in $k*n^2$
  - No inter-domain
- Complex configuration
  - Tunnel interfaces
- Complex steering
  - PBR, autoroute
SR TE

• Simple, Automated and Scalable
  • No core state: state in the packet header
  • No tunnel interface: “SR Policy”
  • No headend a-priori configuration: on-demand policy instantiation
  • No headend a-priori steering: on-demand steering

• Multi-Domain
  • XTC for compute
  • BSID for scale

• Lots of Functionality
  • Designed with lead operators along their use-cases
SR Policy
SR Policy

segment-routing
traffic-eng
policy FOO
end-point ipv4 1.1.1.4 color 20
binding-sid mpls 1000
path
  preference 100
  explicit SIDLIST1
preference 200
dynamic mpls
metric
type latency
affinity
exclude-any red

explicit-path name SIDLIST1
  index 10 mpls label 16002
  index 20 mpls label 30203
  index 30 mpls label 16004

SR policy (1.1.1.4, 20)
Path received via BGP signaling
  preference 300
  binding-sid mpls 1000
  weight 1, SID list <16002, 16005>
  weight 2, SID list <16004, 16008>
Path received via PCEP signaling
  preference 400
  binding-sid mpls 1000
  SID list <16002, 16005>
Path received via NETCONF signaling
  preference 500
  binding-sid mpls 1000
  SID list <16002, 16005>

FIB @ headend
Incoming label: 1000
Action: pop and push <16002, 30203, 14004>
SR Policy

• An SR Policy is identified through the following tuple:
  – The head-end where the policy is instantiated/implemented
  – The endpoint (i.e.: the destination of the policy)
  – The color (an arbitrary numerical value)

• At a given head-end, an SR Policy is fully identified by the <color, endpoint> tuple

• An endpoint can be specified as an IPv4 or IPv6 address
SRTE DB

• A headend can learn an attached domain topology via its IGP or a BGP-LS session

• A headend can learn a non-attached domain topology via a BGP-LS session

• A headend collects all these topologies in the SR-TE database (SRTE-DB).

• The SRTE-DB is multi-domain capable
Path’s source does not influence selection

Selection depends on validity and best (lowest preference)

Provided by e.g. local configuration

Provided by e.g. BGP
Path’s source does not influence selection

Selection depends on validity and best (lowest preference)

Provided by e.g. local configuration

Provided by e.g. BGP SRTE

SR Policy

Path 1
- Pref 100
  - Weight 1
    - SID-list 11: \{16003, 16004\}
    - SID-list 12: \{16004\}

Path 2
- Pref 110
  - Weight 4
    - SID-list 21: \{16004\}

Path 3
- Pref 10
  - INVALID
    - SID-list 31: \{16003, 16004\}
Dynamic Path
Headend Computation
Prefer SR-native Algorithm

Classic Circuit Algo is not optimum!
SID List: {4, 5, 7, 3}
Poor ECMP, big SR list
ATM optimized

SR-native is optimum
Shortest SID list with Max ECMP
SID List: {7, 3}
IP-optimized
Min-Metric with Margin

Min-Metric(1 to 3, TE)
= SID-list <16005, 16004, 16003>
Cumulated TE metric = 23

Min-Metric(1 to 3, TE, m=5, s<=6)
= SID-list <16005, 16003>
Max Cumulated TE metric = 25 < 23+ 5
segment-routing
traffic-eng
policy FOO
  end-point 1.2.3.4 color 10
  path
    preference 100
    affinity
      include-any RED
      exclude-any BLACK
    address
      include PFXSET1
      exclude PFXSET2
    srlg
      include 123
      exclude 654
  admin-tag
    include 1111
    exclude 3333
!
  dynamic mpls
    metric
      type te
      limit 200
    sid-limit 5
    sid-list-limit 1
  association group 1

type node

headend computes a SID list respecting these constraints
Low-Latency

SID-list: {16005, 16004, 16003}

- Min-metric on TE metric where propagation latency is encoded in TE metric
  - same with margin and Max-SID
  - same with latency metric automatically measured by a node for its attached links and distributed in the IGP
Plane Affinity

- Min-Metric on IGP metric with exclusion of a TE-affinity “Plane2”
  - all the links part of plane 2 are set with TE-affinity “Plane2”
Service Disjointness from same headend

- The headend computes two disjoint paths

SID-list: {16002, 30203, 16007}
SID-list: {16005, 16007}

Default IGP link metric: I:10

segment-routing
traffic-eng
policy POLICY1
  end-point ipv4 1.1.1.7 color 100
  path
    preference 50
    dynamic mpls
    metric
type igp
association group 1 type node

policy POLICY2
  end-point ipv4 1.1.1.7 color 200
  path
    preference 50
    dynamic mpls
    metric
type igp
association group 1 type node
On-demand SR Policy

Intra-Domain
On-Demand SR Policy

- A service head-end **automatically instantiates** an SR Policy to a BGP nhop when required (on-demand), **automatically steering** the BGP traffic into this SR Policy

- Color community is used as SLA indicator

- Reminder: an SR policy is defined (endpoint, color)
Different VPNs need different underlay SLA

Basic VPN should use lowest cost underlay path

Premium VPN should use lowest latency path

Objective: operationalize this service for simplicity, scale and performance
On-demand SR Policy work-flow

route-policy ON_DEMAND_SR
  if community matches-any (100:1) then
    set mpls traffic-eng attributeset LOWLAT
  endif
pass
end-policy
!
router bgp 1
neighbor 1.1.1.10
  address-family vpnv4 unicast
    route-policy ON_DEMAND_SR in
  !
segment-routing
  traffic-eng
    attribute-set LOWLAT
    metric
    type te

① 20/8 via CE
  VPN-LABEL: 99999
  Low-latency (100:1)

② 20/8 via PE4
  VPN-LABEL: 99999
  Low-latency (100:1)

③ 20/8 via PE4
  VPN-LABEL: 99999
  Low-latency (100:1)

④ PE4 with Low-latency (100:1)?

⑤ use LOWLAT template

⑥ → SID-list
  {16002, 30204}
Automated performant steering

8 FIB table at PE1
BGP: 20/8 via 4001
SRTE: 4001: Push {16002, 30204}

Automatically, the service route resolves on the Binding SID (4001) of the SR Policy it requires

Simplicity and Performance

No complex PBR to configure, no PBR performance tax

3 20/8 via PE4
VPN-LABEL: 99999
Low-latency (100:1)

2 20/8 via PE4
VPN-LABEL: 99999
Low-latency (100:1)

PE4 with Low-latency (100:1)?

use LOWLAT template

SID-list
{16002, 30204}

7 instantiate SR Policy
BSID 4001

8 forward 20/8
via BSID 4001

FIB table at PE1
Benefits

• **SLA-aware BGP service**

• **No** a-priori full-mesh of SR policy configuration
  – 3 to 4 common optimization templates are used throughout the network
    > color => optimization objective

• **No** complex steering configuration
  – Automated steering of BGP routes on the right SLA path
  – Data plane performant
  – BGP PIC FRR data plane protection is preserved
  – BGP NHT fast control plane convergence is preserved
• Multi-domain topology
  – Realtime reactive feed via BGP-LS

• Multi-domain path compute with TE optimization and constraint
  – SRTE algorithms

pce
address ipv4 1.1.1.3
Stateful

- XTC remembers the request and updates the SID list upon any topology change
  - Anycast SID’s and Local FRR (TILFA) minimize traffic loss during the stateful re-optimization
• We leverage well-known standardized PCE HA
Fundamentally Distributed

- XTC not to be considered as a single “god” box
- XTC is closer to RR
- Different vPE’s can use different pairs of XTC’s
- XTC preference can either be based on proximity or service
Service Disjointness

- Two dynamic paths between two different pairs of (headend, endpoint) must be disjoint from each other.
There is no a-priori route distribution between domains
Inter-Domain Path – Low-Latency

- There is no a-priori route distribution between domains
- An end-to-end policy is requested

SID-list: {30102, 30203}

Segment-routing
traffic-eng
policy POLICY1
end-point ipv4 1.1.1.3 color 20
path
preference 50
dynamic mpls pce
metric
type te
On-demand Next-hop

Inter-Domain
Inter-Domain Routing

- WAN Aggs are re-distributed down to Metro and DC routing areas
- Nothing is redistributed up

How does vPE1 reaches vPE2?
SR Path Compute Element (PCE)

- Multi-Domain topology
  - **Real-time** reactive feed via BGP-LS/ISIS/OSPF from *multiple domains*
  - Including IP address and SID

- Compute
  - **Stateful** with *native* SRTE algorithms
Service Provisioning

vPE1 learns about a service route with next-hop vPE2

How does vPE1 reach the next-hop?
✓ vPE1 only has routes within DC A1 and to the AGG’s of the WAN domain
✓ Solution: On-Demand Next Hop
On-Demand SR Next-Hop
Reachability

- vPE1’s ODN functionality automatically requests a solution from SR-PCE
- **Scalable** - vPE1 only gets the inter-domain paths that it needs
- **Simple** - no BGP3107 pushing all routes everywhere
On-Demand SR Next-Hop

End-to-End Policy

- vPE1’s ODN functionality automatically requests a solution from SR-PCE
- **Scalable** - vPE1 only gets the inter-domain paths that it needs
- **Simple** - no BGP3107 pushing all routes everywhere

---

1: V via vPE2
VPN-LABEL: 99999, Low-Latency

2: V via vPE2
VPN-LABEL: 99999, Low-Latency

3: vPE2 with Low-latency
ODN config at PE1

route-policy ON_DEMAND_SR
    if community matches-any (100:1) then
        set mpls traffic-eng attributeset LOWLAT
    endif
    pass
end-policy
!
router bgp 1
    neighbor 1.1.1.10
        address-family vpnv4 unicast
            route-policy ON_DEMAND_SR in
!
segment-routing
    traffic-eng
        attribute-set LOWLAT
            pce
        metric
            type te
Conclusion
SR TE

• Simple, Automated and Scalable
  – No core state: state in the packet header
  – No tunnel interface: “SR Policy”
  – No headend a-priori configuration: on-demand policy instantiation
  – No headend a-priori steering: on-demand steering

• Multi-Domain
  – XTC

• Lots of Functionality
  – Designed with lead operators along their use-cases
Your Time Is Now
SRv6

Clarence Filsfils
Cisco Fellow - cf@cisco.com

BRKRST-3122
IPv6 adoption is a reality

Global IPv6 traffic grew 243% in 2015

Globally IPv6 traffic will grow 16-fold from 2015 to 2020

IPv6 will be 34% of total Internet traffic in 2020

% Web pages available over IPv6

Source: 6lab.cisco.com – World maps – Web content
IPv6 provides reachability

Support 5G growth
IPv6 addresses summarization

Micro-services
Support container adoption for micro-services

IPv6

Source Address
Destination Address
IPv6

Metro/Core Network
Legacy DC

Next-Gen Data Center

IoT services

xDSL FTTH
Cable

5G

4G

5G

5G

5G

Legacy DC

5G

FTTH

5G

Cable
SRv6 – Segment Routing & IPv6

- Simplicity
  - Protocol elimination
- SLA
  - FRR and TE
- Overlay
- NFV
- SDN
  - SR is de-facto SDN architecture
- 5G Slicing

SRv6 for anything else

IPv6 for reach
SRv6 for underlay

RSVP for FRR/TE → Horrendous states scaling in k*N^2
IPv6 for reach
Opportunity for further simplification

- NSH for NFV
- Additional Protocol and State
- UDP+VxLAN Overlay
- Additional Protocol just for tenant ID
- SRv6 for Underlay
- Simplification, FRR, TE, SDN
- IPv6 for reach

• Multiplicity of protocols and states hinder network economics
Our commitment to Lead Operators

- Clear track record for SR team
SR for anything
Network as a Computer
Network instruction

- 128-bit SRv6 SID
  - Locator: routed to the node performing the function
  - Function: any possible function (optional argument)
    either local to NPU or app in VM/Container
  - Flexible bit-length selection
Network instruction

• 128-bit SRv6 SID
  – Locator: routed to the node performing the function
  – Function: any possible function (optional argument)
    either local to NPU or app in VM/Container
  – Flexible bit-length selection
Network instruction

- 128-bit SRv6 SID
  - Locator: routed to the node performing the function
  - Function: any possible function (optional argument)
    either local to NPU or app in VM/Container
  - Flexible bit-length selection
Network Program

Next Segment

Locator 1  Function 1
Locator 2  Function 2
Locator 3  Function 3
Argument shared between functions

“Global” Argument

Metadata TLV
SR Header

Segments Left
- Locator 1: Function 1
- Locator 2: Function 2
- Locator 3: Function 3

Metadata TLV
SRv6 for anything

Segments Left

Locator 1  Function 1
Locator 2  Function 2
Locator 3  Function 3

Metadata TLV

Turing
SRv6 for anything

- Optimized for HW processing
  e.g. Underlay & Tenant use-cases

- Optimized for SW processing
  e.g. NFV, Container, Micro-Service

Segments Left

- Locator 1: Function 1
- Locator 2: Function 2
- Locator 3: Function 3

Metadata TLV
IPv6 Header

• Next Header (NH)
  – Indicates what comes next
NH = IPv4

<table>
<thead>
<tr>
<th>Version</th>
<th>Traffic Class</th>
<th>Flow Label</th>
<th>Payload Length</th>
<th>Hop Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Source Address

Destination Address

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>Type of Service</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
<th>Fragment Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time to Live</th>
<th>Protocol</th>
<th>Header Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Options</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NH = IPv6
NH = TCP
NH = UDP

<table>
<thead>
<tr>
<th>Version</th>
<th>Traffic Class</th>
<th>Flow Label</th>
<th>Payload Length</th>
<th>Hop Limit</th>
<th>Source Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Source Port</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Length</td>
<td></td>
<td>Destination Port</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Checksum</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>data</td>
</tr>
</tbody>
</table>
NH = Routing Extension

- Generic routing extension header
  - Defined in RFC 2460
  - Next Header: UDP, TCP, IPv6…
  - Hdr Ext Len: Any IPv6 device can skip this header
  - Segments Left: Ignore extension header if equal to 0

- Routing Type field:
  > 0  Source Route (deprecated since 2007)
  > 1  Nimrod (deprecated since 2009)
  > 2  Mobility (RFC 6275)
  > 3  RPL Source Route (RFC 6554)
  > 4  Segment Routing
NH = SRv6

- NH = 43, Type = 4
SRH

- SRH contains
  - the list of segments
  - Segments left (SL)
  - Flags
  - TLV

- Active segment is in the IPv6 DA
- Next segment is at index SL-1
- The last segment is at index 0
  - Reversed order
SRH Processing
Source Node

• Source node is SR-capable
• SR Header (SRH) is created with
  – Segment list in reversed order of the path
    > Segment List [ 0 ] is the LAST segment
    > Segment List [ n – 1 ] is the FIRST segment
  – Segments Left is set to n – 1
  – First Segment is set to n – 1
• IP DA is set to the first segment
• Packet is send according to the IP DA
  – Normal IPv6 forwarding
Non-SR Transit Node

- Plain IPv6 forwarding
- Solely based on IPv6 DA
- No SRH inspection or update
SR Segment Endpoints

- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    - Decrement Segments Left (-1)
    - Update DA with Segment List [Segments Left]
    - Forward according to the new IP DA
SR Segment Endpoints

- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    > Decrement Segments Left ( -1 )
    > Update DA with Segment List [ Segments Left ]
    > Forward according to the new IP DA
  - ELSE (Segments Left = 0)
    > Remove the IP and SR header
    > Process the payload:
      • Inner IP: Lookup DA and forward
      • TCP / UDP: Send to socket
      • ...

```
<table>
<thead>
<tr>
<th>Version</th>
<th>Traffic Class</th>
<th>Flow Label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
```
<table>
<thead>
<tr>
<th>Payload Length</th>
<th>Next = 43</th>
<th>Hop Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Address = A1::</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Address = A4::</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
```
<table>
<thead>
<tr>
<th>Next Header</th>
<th>Len= 6</th>
<th>Type = 4</th>
<th>SL = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>First = 2</td>
<td>Flags</td>
<td>RESERVED</td>
<td></td>
</tr>
</tbody>
</table>
```
```
| Segment List [ 0 ] = A4:: |
| Segment List [ 1 ] = A3:: |
| Segment List [ 2 ] = A2:: |
```
```
<table>
<thead>
<tr>
<th>Payload</th>
</tr>
</thead>
</table>
```
SR Segment Endpoints

- SR Endpoints: SR-capable nodes whose address is in the IP DA
- SR Endpoints inspect the SRH and do:
  - IF Segments Left > 0, THEN
    > Decrement Segments Left ( -1 )
    > Update DA with Segment List [ Segments Left ]
    > Forward according to the new IP DA
  - ELSE (Segments Left = 0)
    > Remove the IP and SR header
    > Process the payload:
      - Inner IP: Lookup DA and forward
      - TCP / UDP: Send to socket
      - ...

Standard IPv6 processing
The final destination does not have to be SR-capable.
Use-Cases
SID allocation for illustration purpose

• For simplicity
• Node K advertises prefix AK::/64
• The function is encoded in the last 64 bits
Endpoint

- For simplicity
- Function 0 denotes the most basic function
- Shortest-path to the Node
Endpoint then xconnect to neighbor

- For simplicity
- AK::CJ denotes

Shortest-path to the Node K and then x-connect (function C) to the neighbor J
TILFA

• 50msec Protection upon local link, node or SRLG failure

• Simple to operate and understand
  – automatically computed by the router’s IGP process
  – 100% coverage across any topology
  – predictable (backup = postconvergence)

• Optimum backup path
  – leverages the post-convergence path, planned to carry the traffic
  – avoid any intermediate flap via alternate path

• Incremental deployment

• Distributed and Automated Intelligence
TILFA

- **50msec Protection upon** local link, node or SRLG failure
- **Simple to operate and understand**
  - automatically computed by the router’s IGP process
  - 100% coverage across any topology
  - predictable (backup = postconvergence)
- **Optimum backup path**
  - leverages the post-convergence path, planned to carry the traffic
  - avoid any intermediate flap via alternate path
- Incremental deployment
- Distributed and Automated Intelligence
Distributed & Automated TE

- IGP minimizes cost instead of latency

FIB
A2::/64 → OIF MOS
A3::/64 → OIF NY

FIB
A3::/64 → OIF TOK

BGP
Advert X/64
Advert Y/64 with Latency
Distributed & Automated TE

- Distributed and Automated Intelligence
- Dynamic SRTE Policy triggered by learning a BGP route with SLA contract
- No PBR steering complexity, No PBR performance tax, No RSVP, No tunnel to configure

BGP
X/64 → A3::0
Y/64 → A3::0 with Low Latency

FIB
A2::/64 → OIF MOS
A3::/64 → OIF NY
X/64 → A3::0
Y/64 → insert {A2::0, A3::0}

Y/64 via A3::0 Low-Latency
X/64 via A::3::0 along IGP path
Centralized TE

Input Acquisition
- BGP-LS
- Telemetry

Policy Instanciation
- PCEP
- BGP-TE
- Netconfig / Yang

Algorithm
- SR native

Low-Latency to 7 for application ...

Low Lat, Low BW

Default ISIS cost metric: 10

Ctrl
Overlay

- Automated
  - No tunnel to configure

- Simple
  - Protocol elimination

- Efficient
  - SRv6 for everything
Overlay with Underlay Control

- SRv6 does not only eliminate unneeded overlay protocols
- SRv6 solves problems that these protocols cannot solve
Spray

Spray Policy 1: {A2::0, A4::0, M1, DD::}
Spray Policy 2: {A3::0, A5::0, M1, DD::}

Replicate traffic to every CMTS through TE-Engineered core path then to access mcast tree then to anycast TV

Flexible, SLA-enabled and Efficient content injection without multicast core
Integrated NFV

- A3::A32 means
  - App in Container 32
  - @ node A3::/64

- Stateless
  - NSH creates per-chain state in the fabric
  - SR does not

- App is SR aware or not
Integrated NFV

- Integrated with underlay SLA

IPv6: (A1::0, A4::0)
SRH: {A3::A32, A4::0, A5::A76, A2::C4}
IPv6: (T::1, V::1)

payload
Integrated NFV

- **A5::A76 means**
  - App in VM 76
  - @ node A5::/64
- **Stateless**
  - NSH creates per-chain state in the fabric
  - SR does not
- **App is SR aware or not**

![Diagram](image)

- IPv6
  - (A1::0, A5::A76)
- SRH
  - { A3::A32, A4::0, A5::A76, A2::C4 }
- IPv6
  - (T::1, V::1)
- Payload
Integrated NFV

• Integrated with Overlay

IPv6: (A1::0, A2::C4)
SRH: {A3::A32, A4::0, A5::A76, A2::C4}
IPv6: (T::1, V::1)
payload

T/64

Server 3
App 32 Container

Server 5
App 76 VM

IPv6: (T::1, V::1)
payload

V/64
More use-cases

• 6CN: enhancing IP to search for Content
• 6LB: enhancing load-balancers
• Video Pipeline
• 5G Slicing
• 5G Ultra-Low Latency
SRv6 status

• Cisco HW
  – ASR9k - XR
  – ASR1k – XE

• Open-Source
  – Linux 4.10
  – FD.IO
Conclusion
Network Programming

• An SRv6 segment is a function at a node
• An SRv6 segment list is a network program
• The network acts as a large computer
• Integrated use-cases well beyond underlay (TE, FRR)
  – NFV
  – Container networking
  – Efficient content management: Spray, 6CN, 6LB
  – Video pipeline
• Simplification: IPv6+SRv6 only!
SRv6 Leadership

• Bold architecture
• Numerous use-cases
  – FRR, TE, SDN, Overlay with SLA, NFV, Spray, 6CN, 6LB, 5G Slice & LL
• First to demonstrate HW implementation
• First to FCS, field trial and deployment
• Fund university to bring SRv6 in Linux 4.10
• Fund significant SRv6 implementation in FD.IO
• Feel free to join the lead-operator team!
Complete Your Online Session Evaluation

• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Continue Your Education

- Demos in the Cisco campus
- Walk-in Self-Paced Labs
- Lunch & Learn
- Meet the Engineer 1:1 meetings
- Related sessions
Thank You
Complete Your Online Session Evaluation

• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online
Continue Your Education

• Demos in the Cisco campus
• Walk-in Self-Paced Labs
• Lunch & Learn
• Meet the Engineer 1:1 meetings
• Related sessions
Thank You
Your Time Is Now