Exposing Link-Change Events to Applications

Problem Description

As a Mobile Node (MN) handoffs from one Point Of Attachment (POA) to another, the state of its network interface (in the MN) changes:

- From link-up to link-down – when disconnecting from one POA
- From link-down to link-up – when connecting to the new POA

This may also be accompanied by a need to refresh its source IP address

Problem Description - continued

The implications on running applications could be significant:

- Most Socket APIs are blocking – hence the invoking apps are blocked throughout the duration of the link-down state
- Sending TCP data while the link is down might cause time-outs that could be easily avoided if the app had knowledge of the link state and postponed its transmissions until the link is up again
- If the source IP address had changed, it is not clear what happens to the associated active sockets
- In some DMM use-cases, it is desirable that apps restart the socket to enjoy a better service (even when the current service is still available)

Problem Description - continued

Actually, the desired behavior is also influenced by:

- The type of source IP address that was assigned to the MN (Nomadic or Sustained/Fixed)
- The location of the new POA compared to the origin – same or different address prefix
- The network level of support for make-without-break handoff

Relevant RFCs

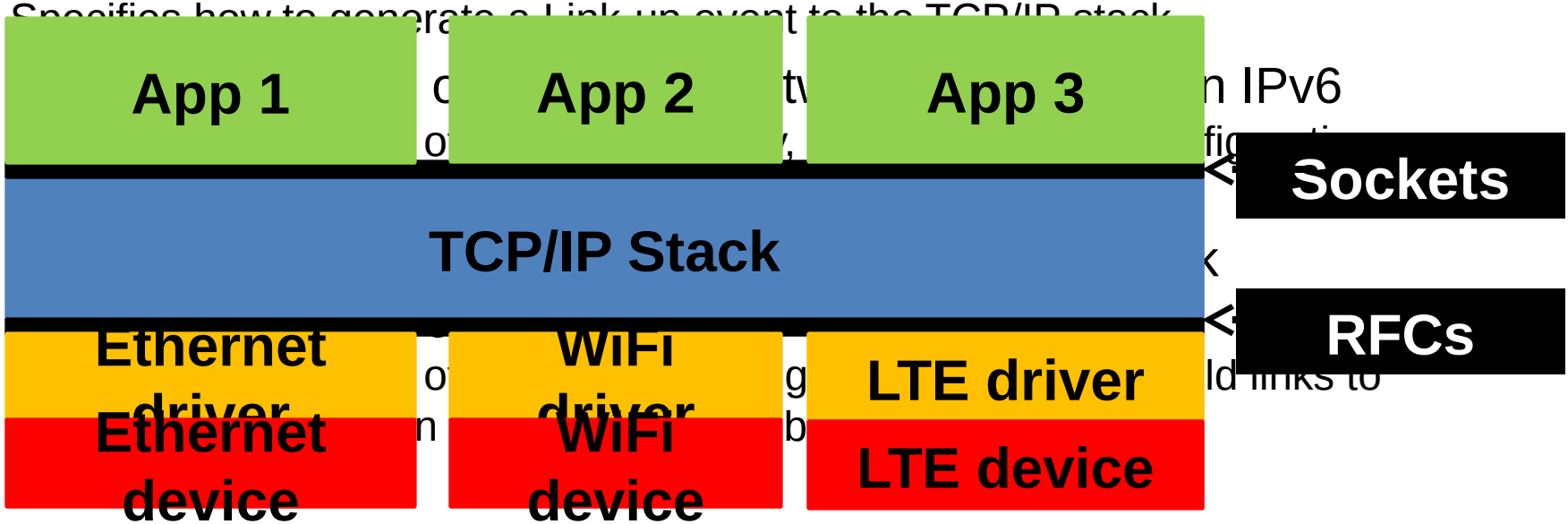
There are some RFCs that discuss link-change related topics:

- RFC4957 – Link-Layer Event Notification for Detecting Network Attachments
  Specifies how to generate a Link-up event to the TCP/IP stack
- RFC4135 – Goals of Detecting Network Attachment in IPv6
  Discusses the benefit of detection. Mainly, triggering new IP configuration when required in minimum latency.
- RFC6059 – Simple Procedures for Detecting Network Attachment in IPv6
  Discusses the benefit of retaining IP configuration attributes of old links to expedite re-connection when handing-off back to that old link

Relevant RFCs

There are some RFCs that discuss link-change related topics:

- RFC4957 – Link-Layer Event Notification for Detecting Network Attachments
  Specifies how to generate a Link-up event to the TCP/IP stack
- n IPv6
- k
- ld links to

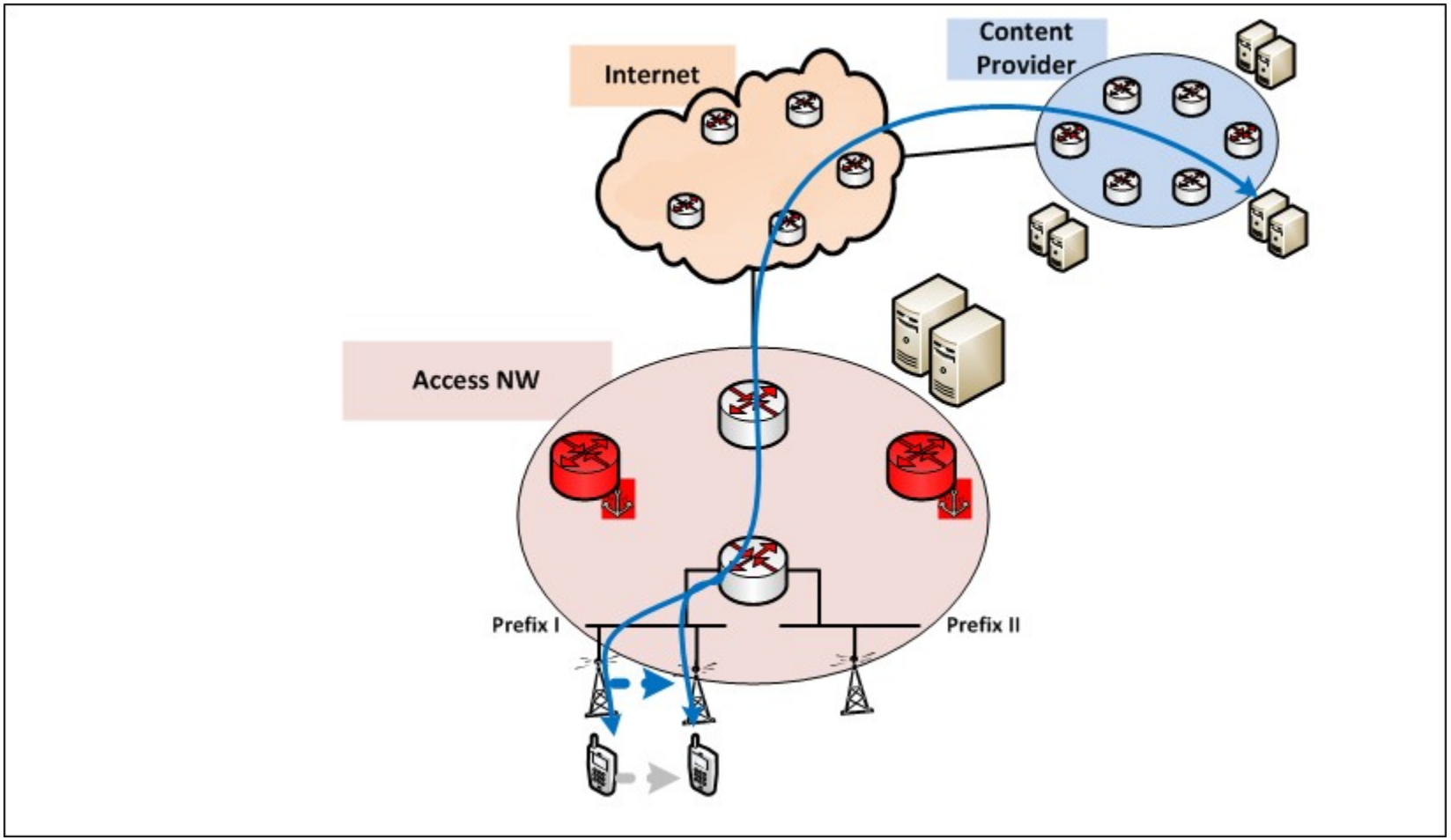


Use-cases

Following are some use-cases to examine:

- Handoff – Nomadic address – no LAN change (same NW prefix)
- Handoff – Nomadic address – new LAN
- Handoff – Sustained address – new LAN
- Handoff – Sustained address – new LAN – better service available
- Link-down due to error
- Handoff – Nomadic address – CSIPTO support

Handoff – Nomadic address – no LAN change



Handoff – Nomadic address – no LAN change

Since the LAN did not change, there is no change in the source IP address.

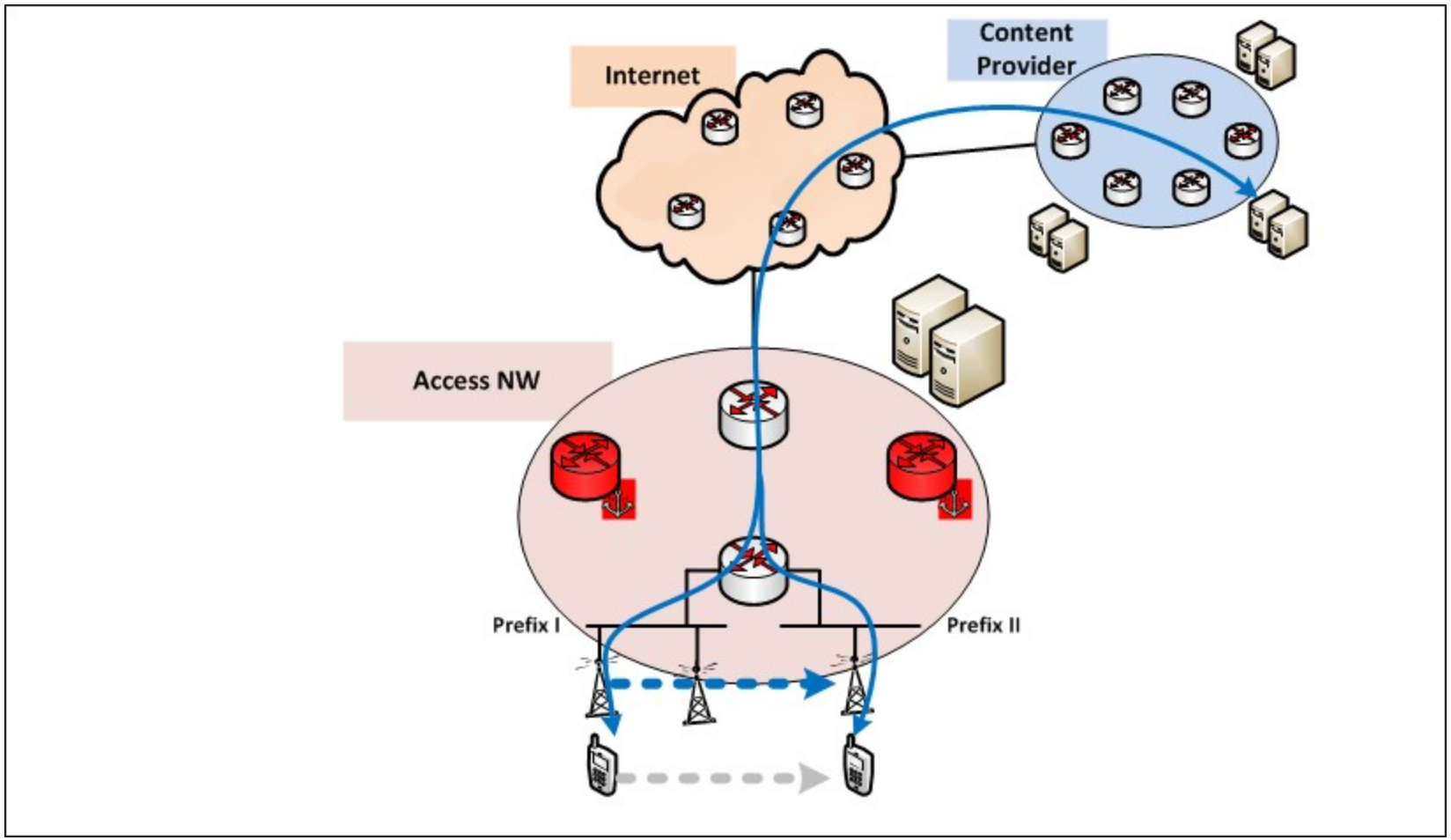
Preferred behavior:

If the handoff is quick – no event should be triggered

If the handoff is slow – better trigger a link-down and link-up events:

- **Upon link-down** –apps should not attempt to transmit data
- **Upon link-up** - apps should resume normal operation

Handoff – Nomadic address – new LAN



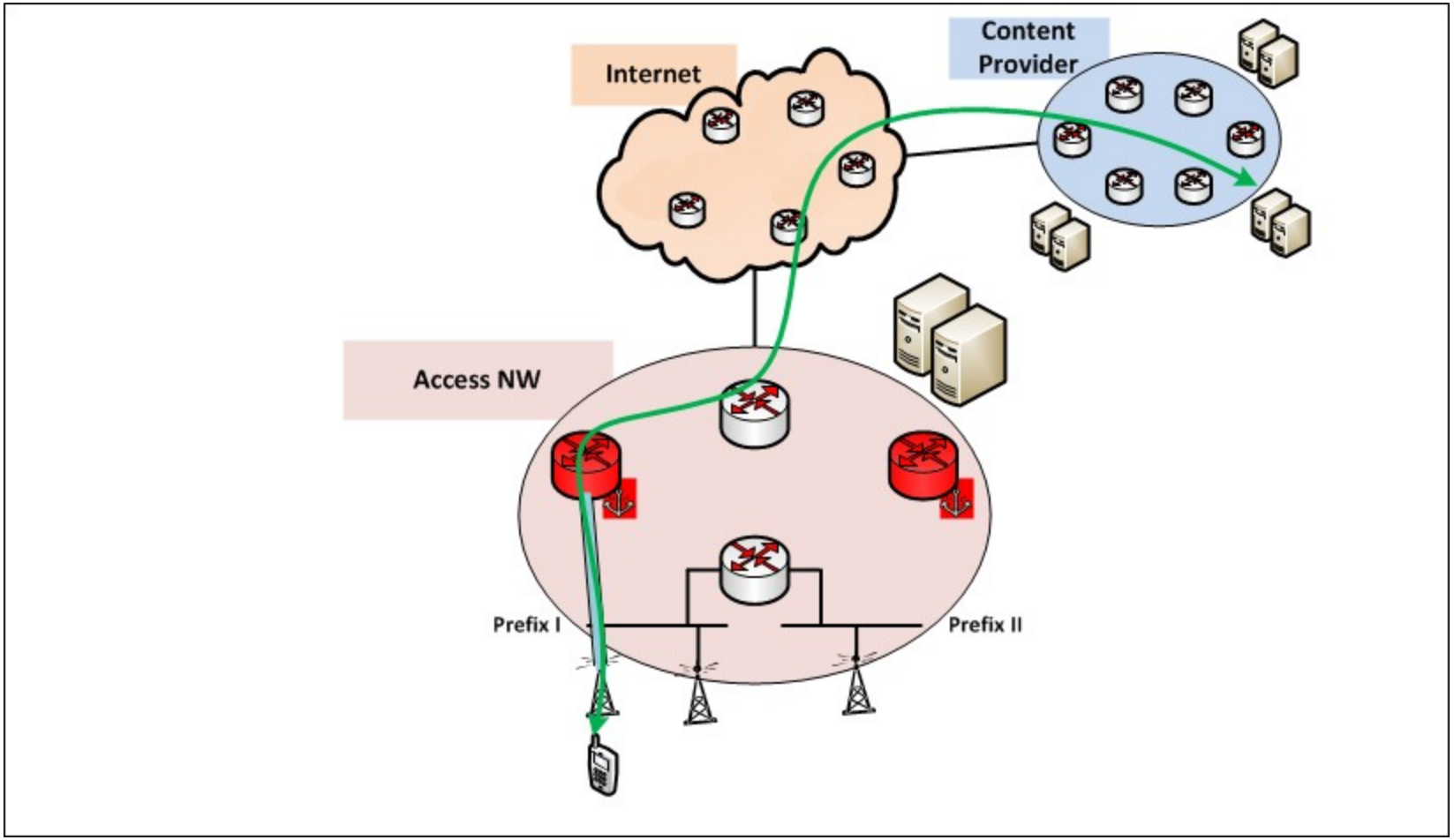
Handoff – Nomadic address – new LAN

New LAN means a different network prefix – so a new nomadic IP address should be obtained.

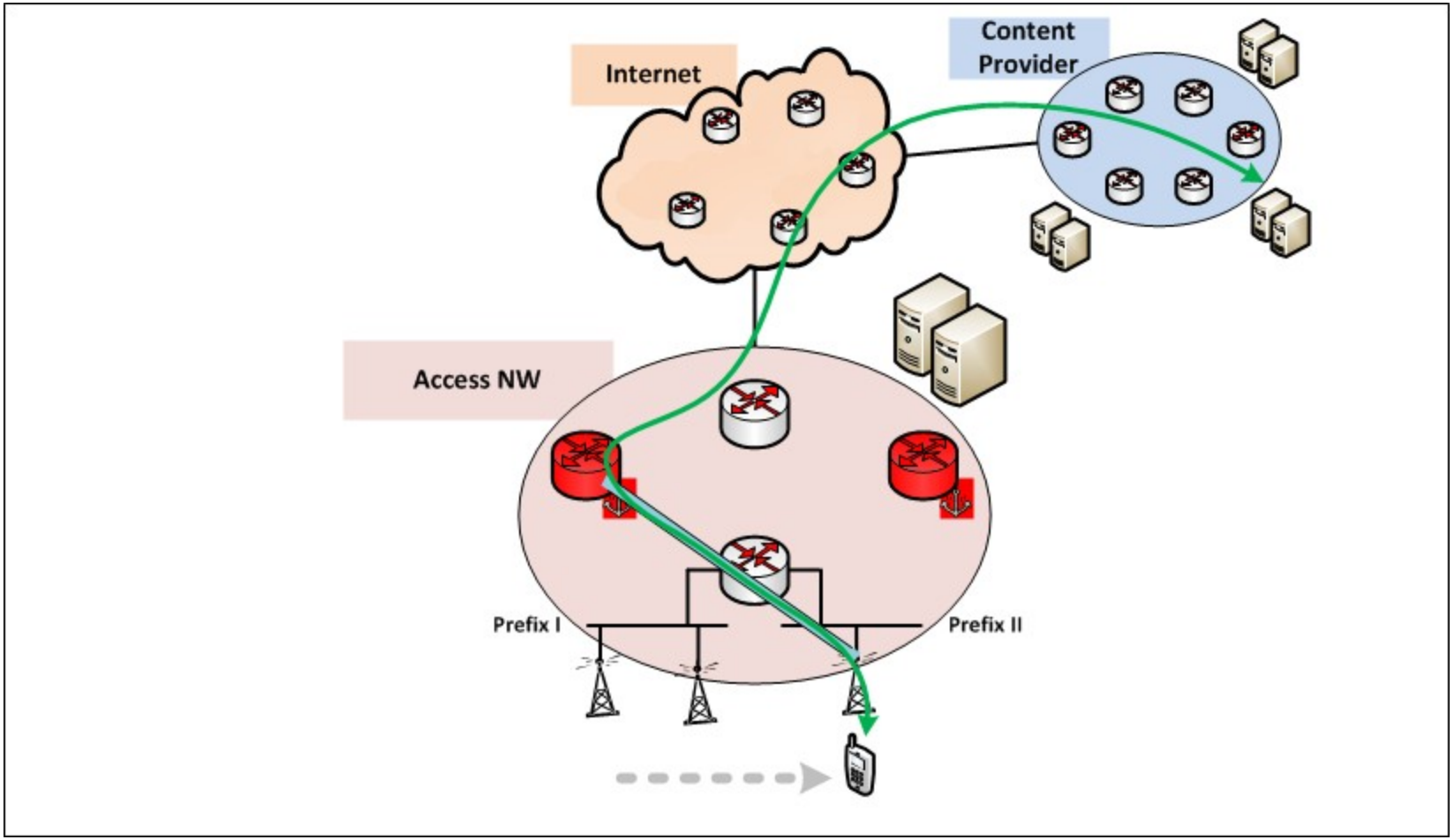
Preferred behavior:

- **Upon link-down** – apps should not attempt to transmit data
- **Upon link-up** – apps should close the socket associated with the obsolete IP address and create a new socket prior to resuming to normal operation

Handoff – Sustained address – new LAN



Handoff – Sustained address – new LAN



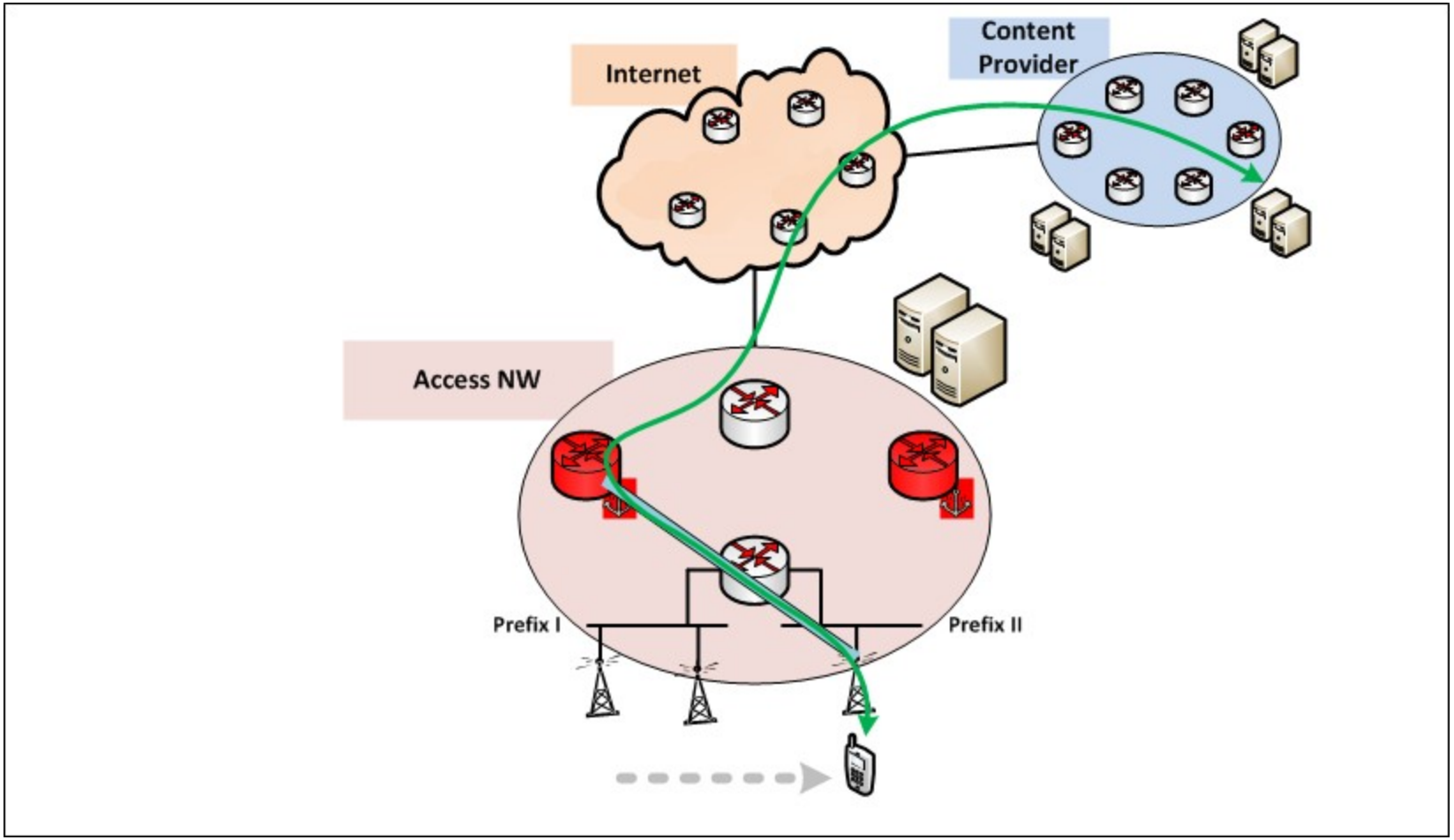
Handoff – Sustained address – new LAN

Sustained address means that the network guarantees IP session continuity.

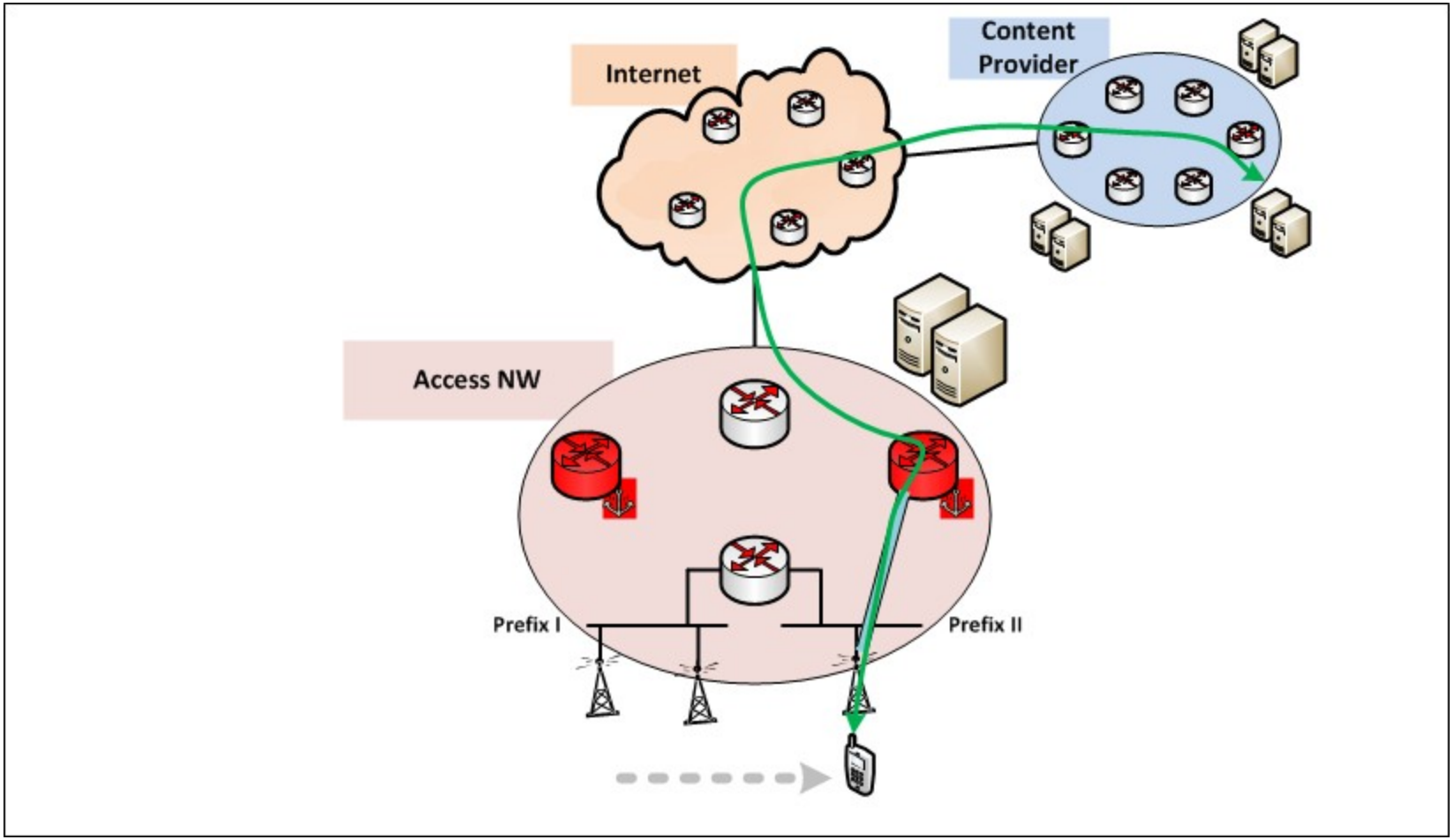
Preferred behavior:

- **Upon link-down** – apps should not attempt to transmit data
- **Upon link-up** – Apps may resume normal operation (the source IP address and socket are still valid)

Handoff – Sustained address – new LAN – better service available



Handoff – Sustained address – new LAN – better service available



Handoff – Sustained address – new LAN – better service available

Sustained address means that the network guarantees IP session continuity.

Better route through a different Mobility Anchor

Preferred behavior:

- **Upon link-down** – apps should not attempt to transmit data
- **Upon link-up** –
  - Apps may resume normal operation (the source IP address and socket are still valid)
  - Apps may choose the best time to close the socket and open a new one using a new Sustained IP address with the better route

But how can apps know that a better route is available?

Link-down due to error

The link is down. It is unknown when it will be up again…

Preferred behavior:

Stop performing networking operation

Update the user (if possible)

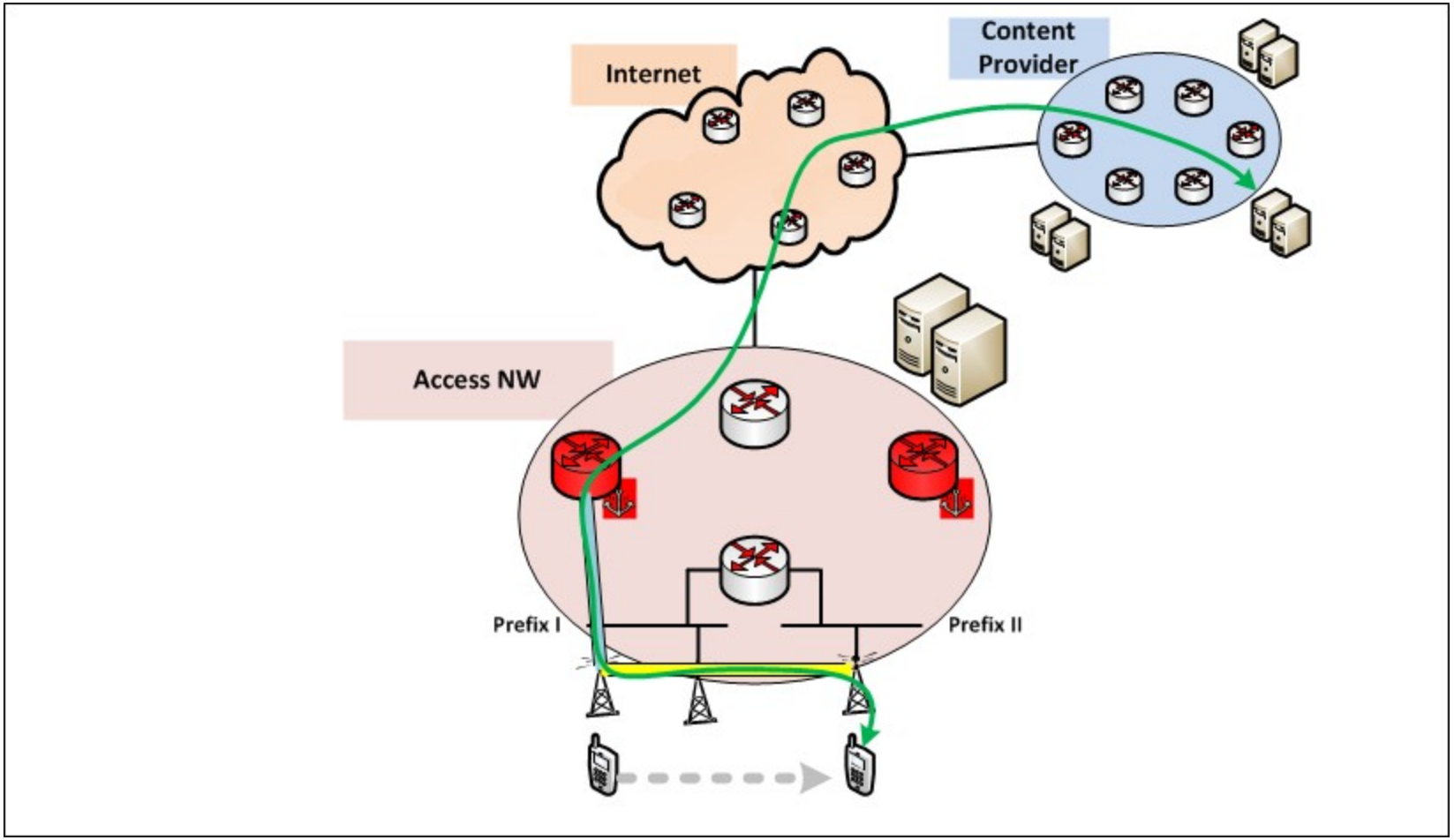
But how can the Socket layer identify the cause of the link-down event as an error rather than a handoff?

Handoff – Nomadic address – CSIPTO support

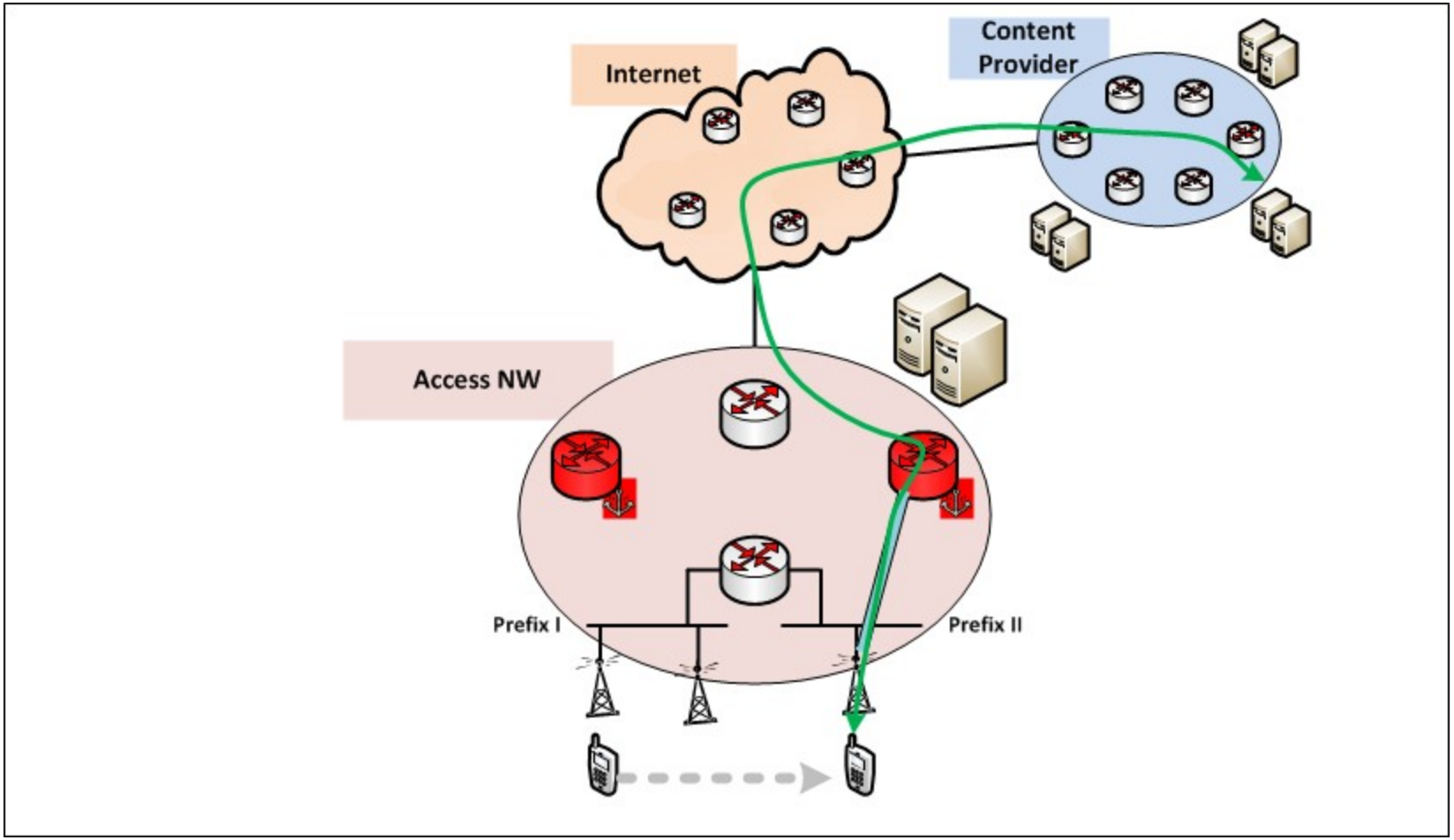
- Mobile access networks with CSIPTO, enable mobile nodes to handoff to a new point-of-access without experiencing a link-down/link-up event.
- The main motivation is to provide make-before-brake user experience
- But after the handoff, sub-optimal routes may exists and using a new IP source address is preferable

*CSIPTO – Coordinated Selected IP Traffic Offload

Handoff – Nomadic address – CSIPTO support



Handoff – Nomadic address – CSIPTO support



Handoff – Nomadic address – CSIPTO support II

- Assuming an indication of a handoff event can be generated –

Preferred behavior:

Upon handoff event –

- Apps may continue normal operation , but –
- It is preferable to find a good opportunity to close and re-open the socket
- This is true even for Nomadic IP addresses since CSIPTO may guarantee session continuity even for them

Summary

There are benefits from enabling apps to be aware and act upon link state changes due to:

- Distributed MAs leading to sub-optimal routes
- The different source IP address types
- New and advanced mobile access network services

Proposing to add support in the Socket interface for these events

Next steps

1. Is this interesting to DMM?
2. Write a requirements document for Socket interface support