Ethernet Network Network Interface: Heavy or Light?

Two choices for an ENNI

Rev. 2

Norman Finn

nfinn@cisco.com

ENNI: Heavy or light?

- There are at least two distinct methods we can pursue for defining an Ether NNI:
- **Heavy**:

  A Buffer Network is built along the lines suggested in new-nfinn-buffer-networks-0310-v01.pdf with an explicit data encapsulation.

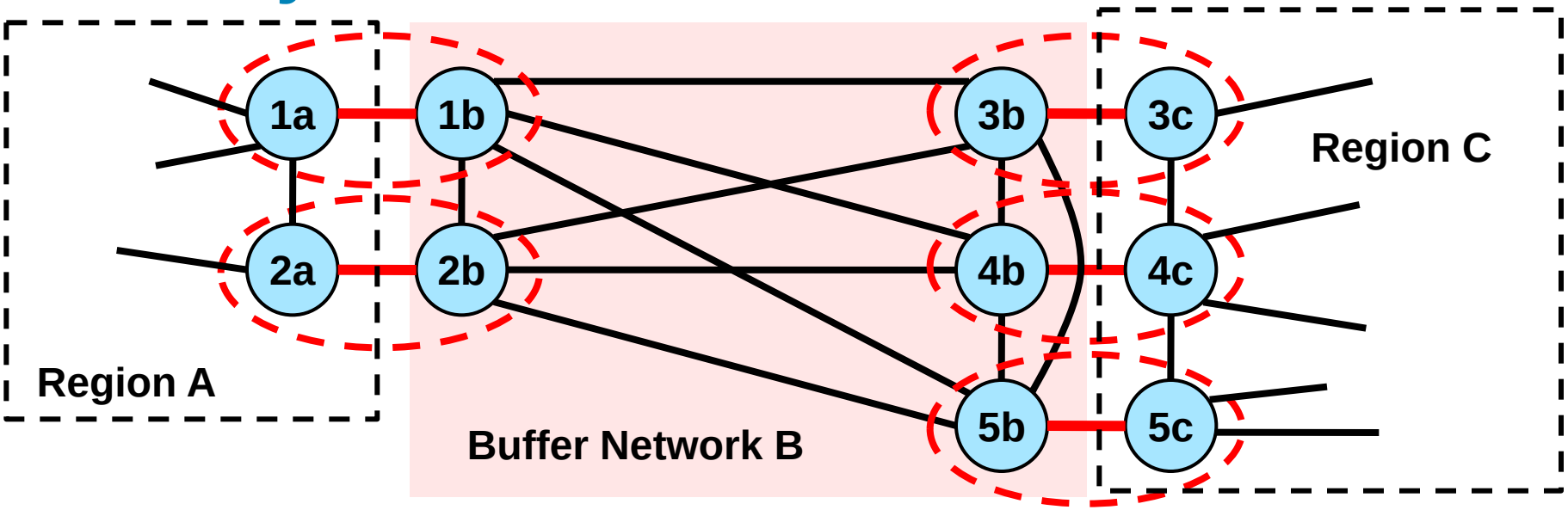
- **Light**:

  Buffer Network is built using “virtual nodes,” i.e. the multiple physical Nodes of each Portal cooperate to give the appearance of a Portal consisting of a single Node.

- Each method has its advantages and drawbacks, and all of the drawbacks can be addressed. This is a classic engineering decision.

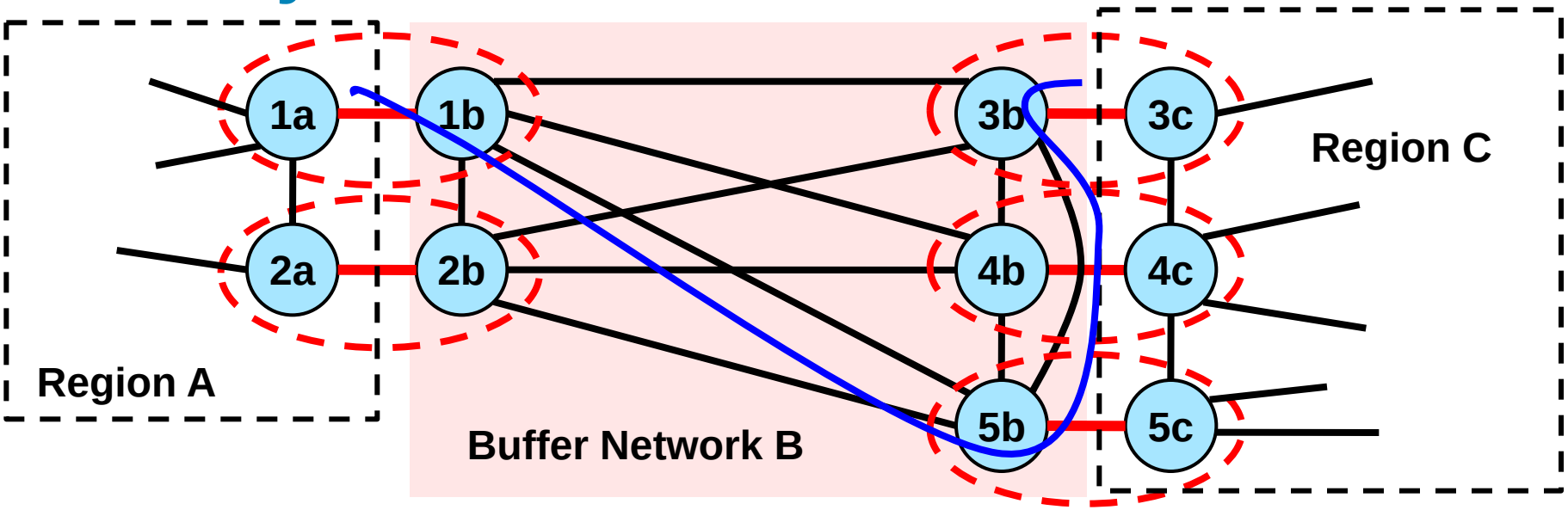
Heavy ENNI

Heavy ENNI



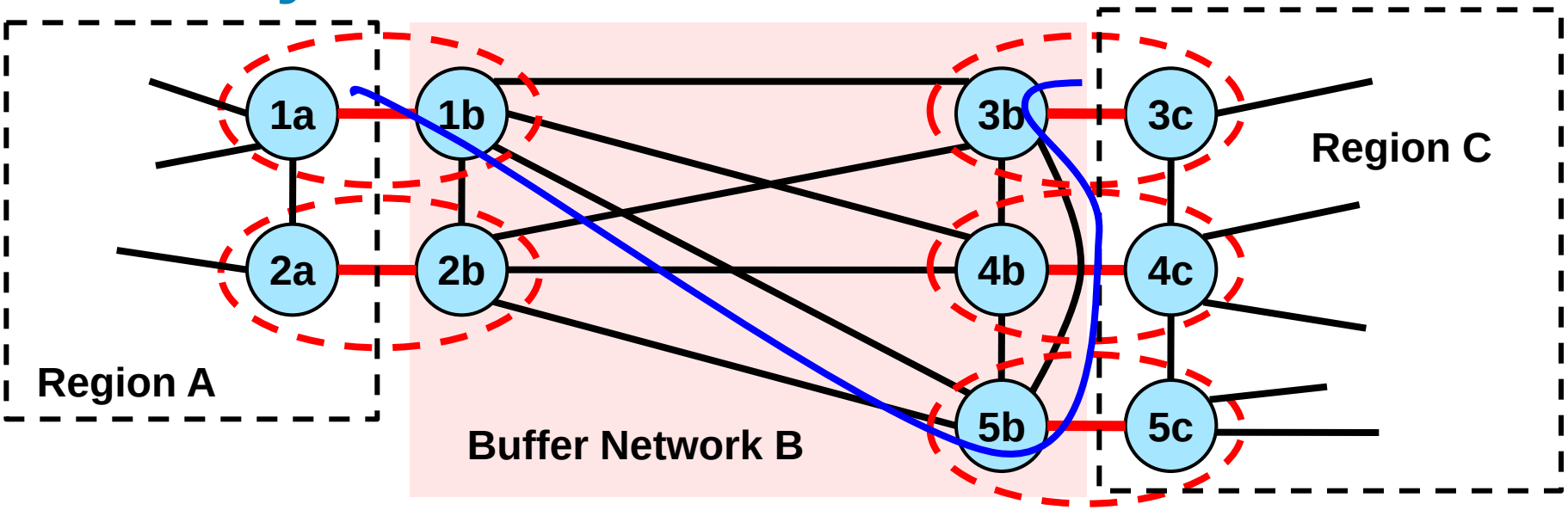
- The Regions choose which NNI (shown by a heavy red line) is used for each service.
- Each Node in the Buffer Network, whether real or virtual, must exist, so that data can be reliably delivered from NNI to NNI.
- One would expect that the Terminal Pairs (1a-1b, 5b-5c, etc.) are physical devices, and the NNIs virtual.

Heavy ENNI



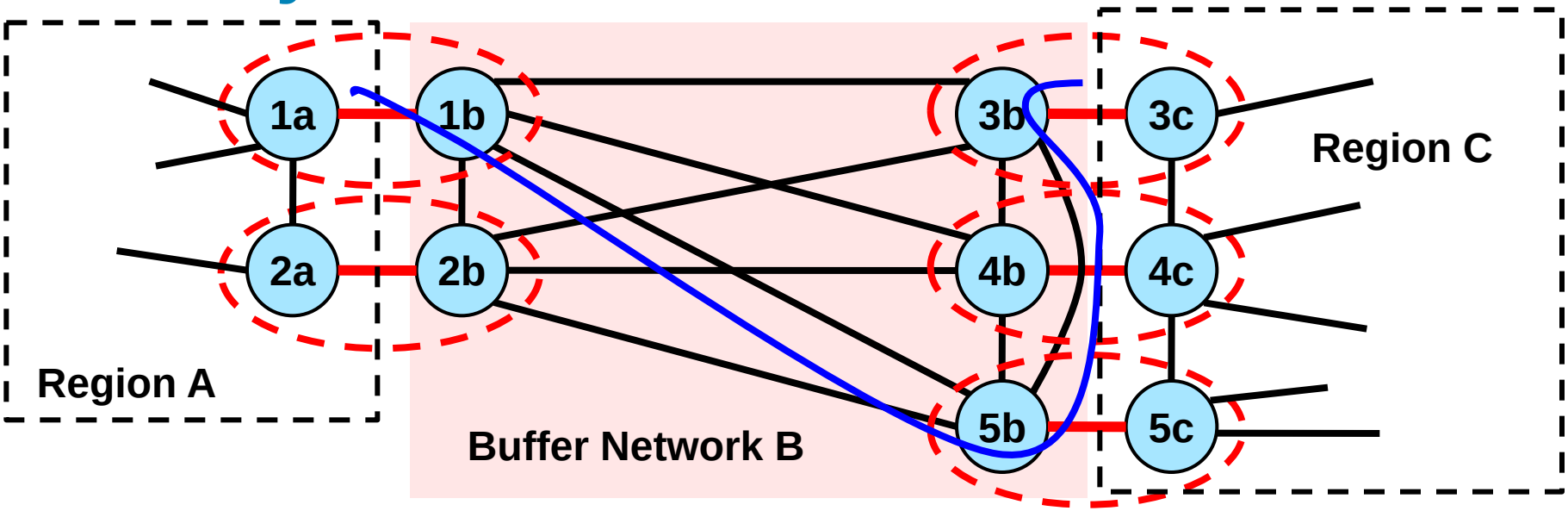
- The Heavy ENNI maximizes the independence of the two Regions.
- In order to minimize failure recovery times, we create explicit paths through the Buffer network from NNI to NNI, e.g. **(3c –) 3b – 5b – 1b (– 1a)**, protected (presumably) by CFM.

Heavy ENNI – issues



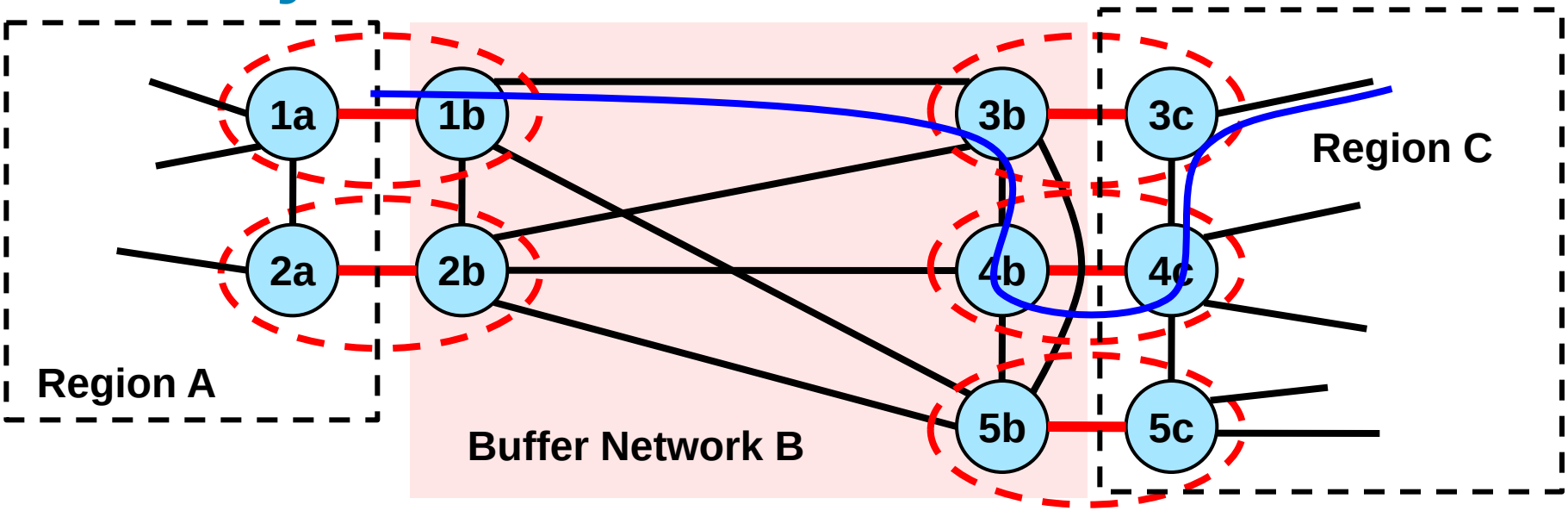
- The physical links between Terminal Nodes (e.g. between 3 and 4) are shared by the Buffer Network (3b-4b) and the Region (3c-4c).
- This makes the choice of encapsulation for frames between 3 and 4 or 1 and 2 somewhat difficult, and can affect the choice of encapsulation across the Buffer Network (1b-3b).

Heavy ENNI – issues



- If the data forwarding and encapsulation methods of the two Regions are different (e.g. 802.1ad vs. VPLS), the choice of encapsulation for the Buffer Network can be difficult, especially with regard to physical link sharing (3-4 or 1-2).
- Separating the Terminal Pairs into physically separate devices solves this, of course, but is expensive.

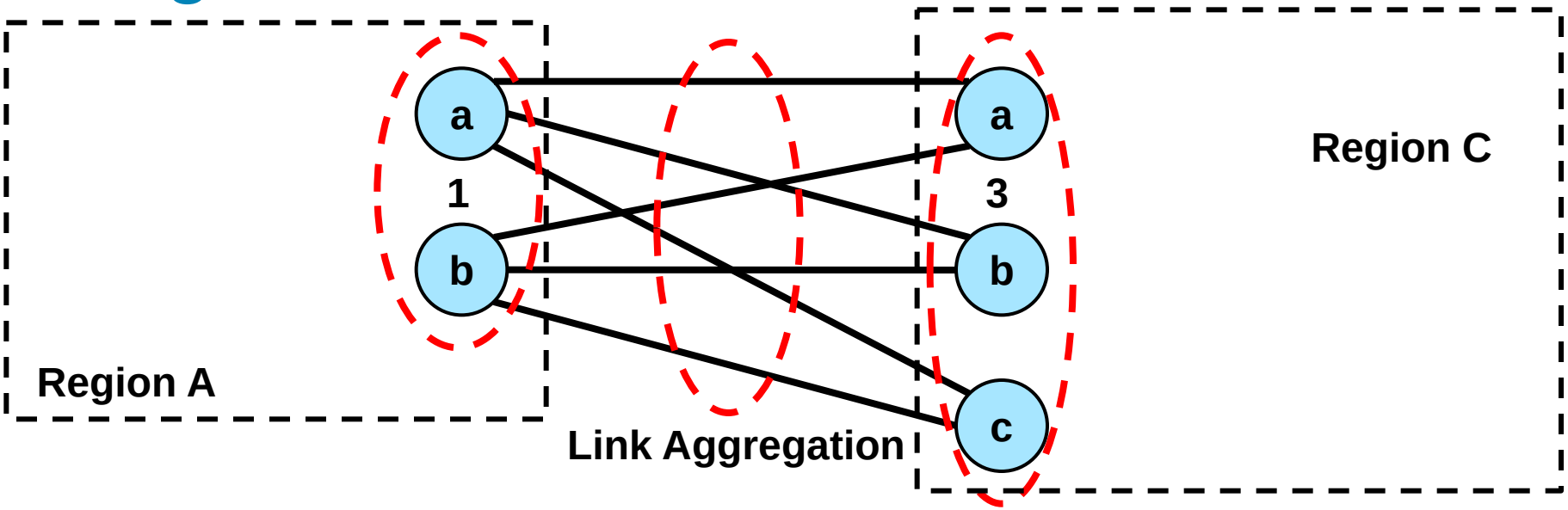
Heavy ENNI – issues



- The need to minimize the propagation of errors by enforcing a single NNI entry/exit point for each service can lead to “hairpinning” data needlessly across a physical link, as above where1a-1b and 4b-4c are the preferred NNIs for the blue service, requiring the double use of the 3b3c-4b4c physical link.

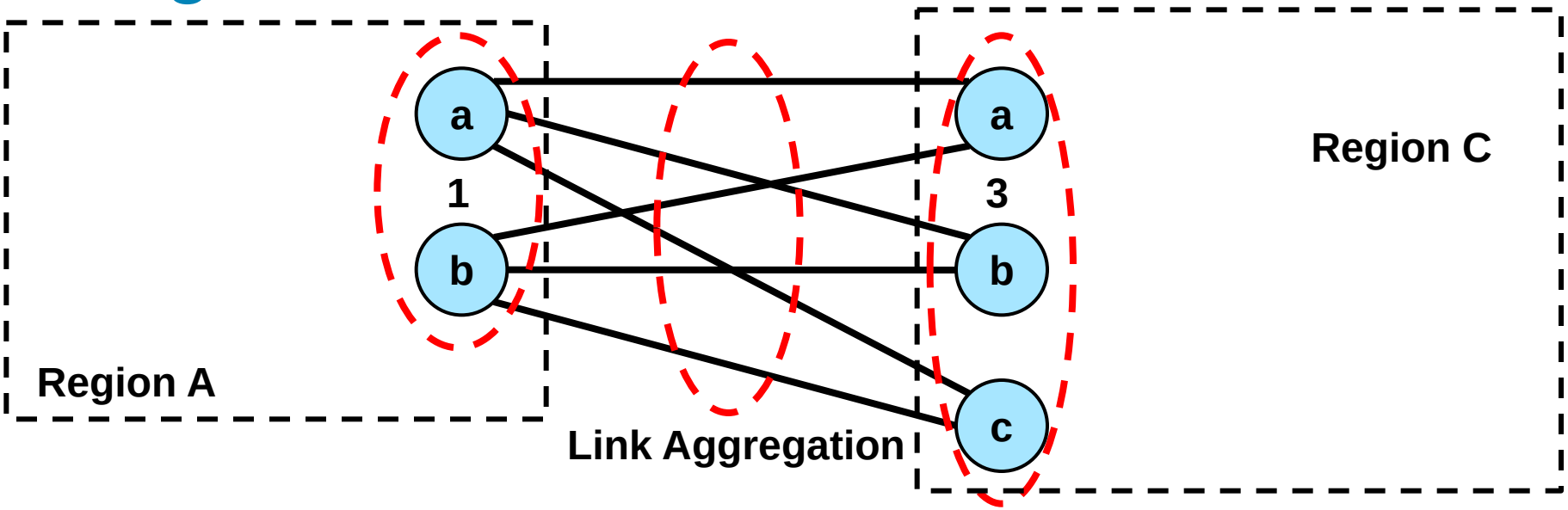
Light ENNI

Light ENNI



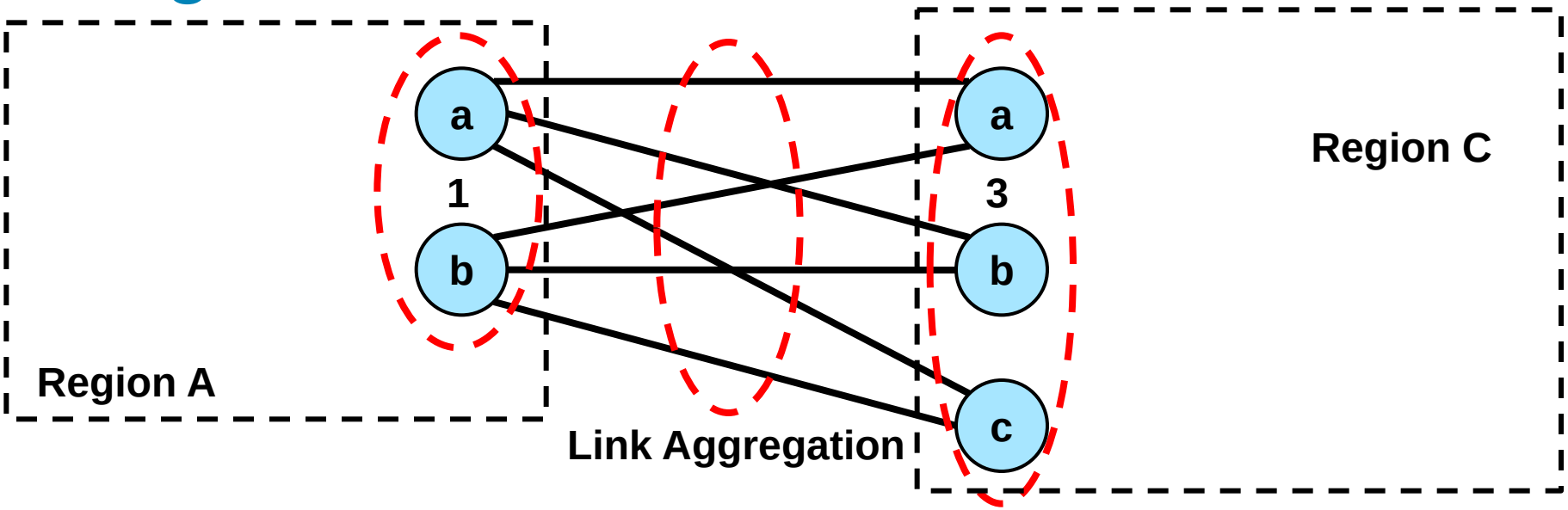
- The Terminal Nodes in each Region appear, to the other Region, to be a single Terminal Node (bridge, switch, or whatever).
- All of the inter-Region Links are combined into a single Aggregated Link using LACP.
- Links among Nodes in the same Region are invisible and irrelevant to the ENNI.

Light ENNI



- The means by which the Virtual Terminal Nodes are implemented does **not** need to be standardized; this author sees no requirement for 3a, 3b, and 3c to come from three different vendors.
- The choice of physical link is always up to the transmitting Virtual Terminal Node, and the receiving Virtual Terminal Node must live with the choice.

Light ENNI

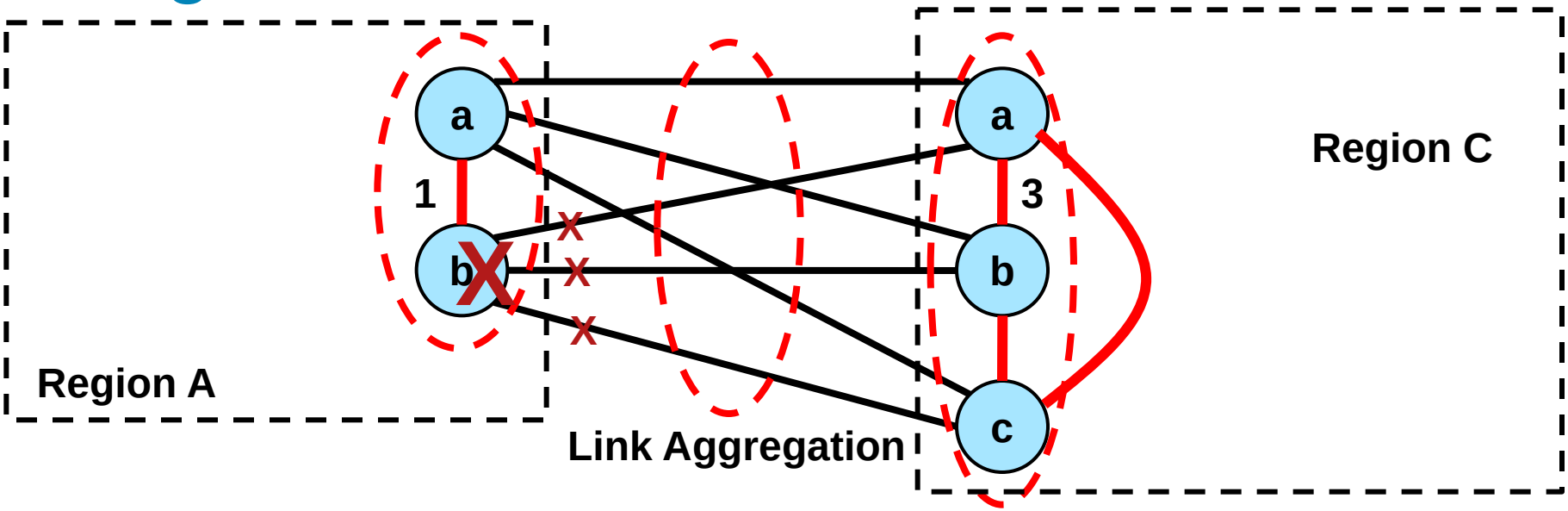


- Physical level CFM can be used to improved failure detection time for the physical links.
- Obviously, the two Regions have to agree on a data encapsulation, but a 1:1 service encapsulation translation can be performed at either (or both) ends, and no encapsulation-dependent CFM is required.

Light ENNI – issues

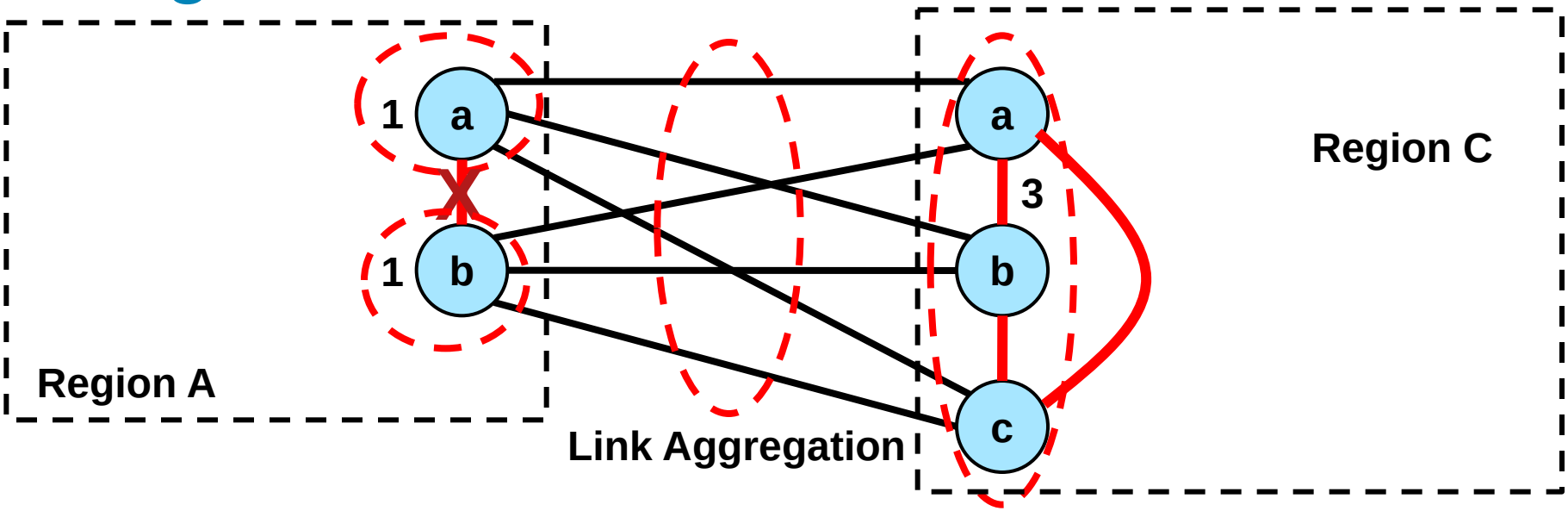
- Clearly, service-based physical link selection is preferred to other methods, e.g., hashing the IP 5-tuple.
- For efficiency of routing, a means (perhaps LACP extensions) should be provided for one Region to express a preference (not a demand) for which link should be used for which service.
- For optimum maintainability, it we should provide a means (perhaps LACP extensions) for the two Virtual Terminal Nodes to agree to use the same physical link for both directions for a given service (or bundle of services).

Light ENNI – issues



- The physical components of a Virtual Terminal Node appear to be a single Node to (at least) the other Region, and perhaps to other Nodes internal to the Region, as well.
- If one component of a Virtual Terminal Node **fails** (say 1b) then its attached Links fail, but the remaining Nodes (1a in this case) continue to function; recovery is quick.

Light ENNI – issues



- If the **Link** between two components of a Virtual Terminal Node (e.g. 1a-1b) fails, both components can takeover the Node's identity, but act independently (the "**split brain**" scenario), with disastrous results.
- For this reason, "inter-VTN links" are made extra-reliable, and in some implementations, are assumed to be failure-proof.

Light ENNI – issues

- We cannot (in the author's opinion) design a network standard around "failure proof links".
- Since we are assuming that LACP is being used to establish Aggregated Links between Virtual Terminal Nodes, we could **enhance LACP** so that the devices connected to a Virtual Terminal Node can assist the VTN in detecting a "split brain" scenario.
- But, split brain detection is necessarily a hippity-hippity-hop operation, involving multiple Nodes; there is no equivalent to the (3c –) 3b – 5b – 1b (– 1a) Maintenance Association described for the Heavy ENNI. Split brain detection will be **slower** than MA failure detection.

Light ENNI – issues

- Recovery from the split brain is up to the implementation:

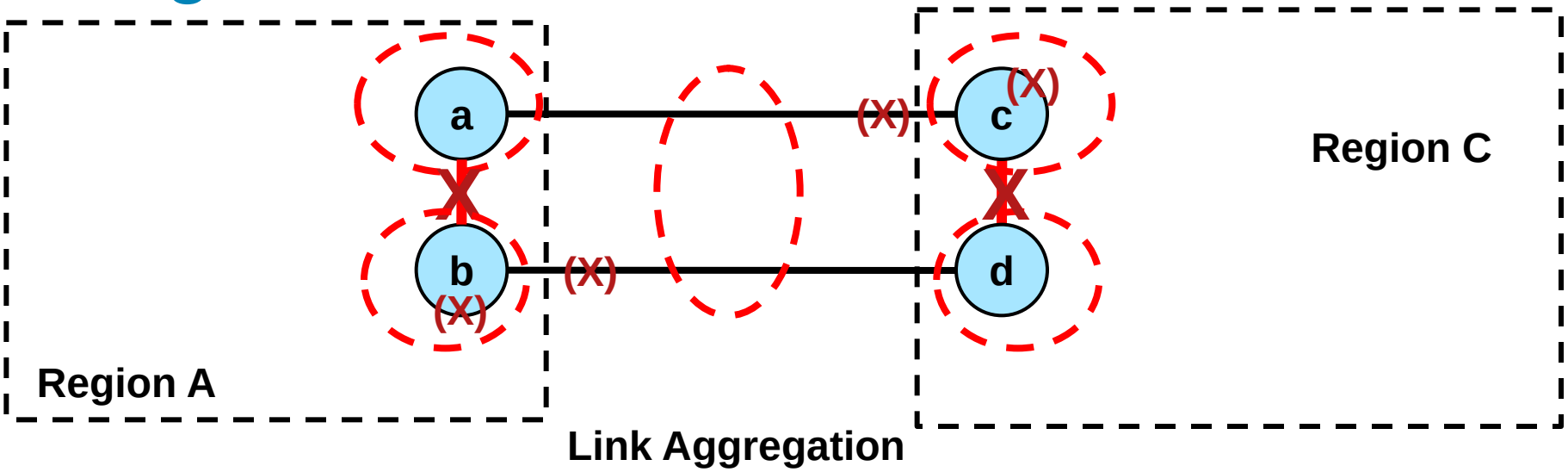
  Some implementations may have no issues with a split brain.

  Some implementations may shut down an isolated secondary component of the virtual node.

  Some implementations may change identities to become two separate devices (equivalent to shut down for the ENNI, since the “light” scheme requires a single virtual node).

- Signaling the recovery choice can be handled with current LACP, e.g., by removing Links to one of the physical Nodes from the aggregation.

Light ENNI – issues



- Suppose the recovery method for “split brain” is that the secondary device shuts down.
- If **a** and **d**, above, are “master” nodes, then if both inter-VTN links fail (as shown), the ENNI would fail.
- Indicating in LACP which is the master node would enable the administrators to make **a** and **c** the master nodes, so that the **a—c** link would remain operational.

Common issues

Common issue: bundling preferences

- We cannot express link preferences for thousands of services in an LACP or CCM PDU; some kind of “bundling” is necessary across the ENNI.
- We can say that the bundling is handled by configuration, and that both administrations must get the configuration right. This seems risky.
- We probably need to define a protocol (or use an existing one) that allows each side to express its bundling preferences, and to tie the LACP or CCM signals to a particular expression.
- We may or may not provide an automatic means of resolving differences in bundling preferences.
- A transport protocol is likely required to carry this much information.

Common issue: service information

- We may or may not provide a means (or use an existing one) of exchanging information about each service passed across the ENNI. Such information could include:

  The existence of a service with a particular service identifier.

  QoS parameters for a service such as EIR/CIR rates, latency requirements, or connectivity priority.

  The global service identifier (used in CCMs) for the service.

  Membership of a VID in a root/individual/group VID set for a rooted multipath service.

  MIRP “I need to receive this service” registrations.

Summary

ENNI: Plusses and minuses

- **Heavy**: intra-Regional connectivity **visible** to ENNI

  + CFM (enhanced) ensures fast failover in all situations.

  – The Buffer Network requires its own encapsulation across Links interior to a Region.

  – Data can be “hairpinned”.

- **Light**: intra-Regional connectivity **invisible** to ENNI

  + Encapsulation choices are flexible; no links are shared between Region and Buffer Network.

  – Detecting a split brain could slow the reaction time when a Link between elements of a virtual Node fails.

- **Common**:

  ? How much more do we provide?

“Light NNI” vs. Alon/Sprecher

Why VTNs and LACP for Light NNI?

- In the data plane, the scheme shown here for “Light NNI” is essentially identical to that given in new-alon-service-protection-over-external-interfaces-03-10-v01-.pdf, which does not talk about virtual nodes or LACP.
- Using virtual nodes and LACP:
  - Sidesteps a number of questions about the data plane, e.g., how can a Region’s routing/bridging protocol ensure delivery over the correct inter-Regional Link?
  - Offers a mechanism (LACP) to provide some assurance that the inter-Regional Links are correctly connected.
  - Provides a basis (again, LACP) for conveying necessary control information such as Service-to-Link assignment preferences.
  - Solves the problem (via LACP), present in both schemes, of distinguishing between Node failures and Link failures.