# Buffer Networks: A Paradigm for Employing NNIs


## Abstract

Separate providers' networks (Regions) are interconnected via Network-Network Interfaces (NNIs) to create networks spanning the globe. To get the fastest reaction to node or link failures, some form of protection switching is common. Because of limitations inherent in the protection switching model, multiregion customer services require either alternative paths through the global network, with path switching performed at the User Network Interfaces (UNIs), or virtual switches emulated by multiple chassis. Both solutions have problems. By using Buffer Network Insertion, by defining a tight set of requirements that can well be met by such Buffer Networks, and by allowing the Regions to determine which NNI to use when transferring between the Region and the Buffer Network, the requirements on the Regions' protection capabilities can be relaxed compared to the Buffer Networks' requirements, the need for end-to-end alternative protected paths for multiregion services can be eliminated, the propagation of fault recovery actions through the network can be avoided.


## 1. Introduction

Providers require interfaces among their Networks, called Network-Network Interfaces (NNIs), so that Services can be sold (jointly or by subcontract) that cannot be offered by any single provider. For example, a Service connecting Paris, Caracas, and San Diego may require the joint efforts of four providers. It has proven difficult to define protocols and procedures for connecting providers' Networks across NNIs while meeting all of the sometimes conflicting requirements and desires of the providers and customers.

The following sections:

a) Provide a list of terms with special meaning in this document (Section 2);
b) Define the problem to be solved (Section 3);
c) Describe two of the most common means for solving the problem (Section 4);
d) Describe the Buffer Network Insertion technique (Section 5) for solving the problem; and
e) List a few references (Section 6).

## 2. Definitions

These definitions may differ somewhat from those in use elsewhere, and in particular, differ from those in [1]. The changes in terminology are shown in Table 1. Figure 1 illustrates these definitions.

**Table 1—Terminology changes**

| new-nfinn-nni-framework-0110-v01.pdf | this document |
|---|---|
| Cloud | Region |
| Network | <not used> |
| Network-Network Interface (NNI) | Buffer Network |
| Node | Node or Terminal Pair |
| Subnetwork | Network |
| Terminus | Network-Network Interface (NNI) |
| Node in a Portal | Terminal Pair |

**2.1 Link**: A physical or logical medium connecting two or more Nodes, with two the most common case. A Link belongs to exactly one Network, the same one as the two Nodes.

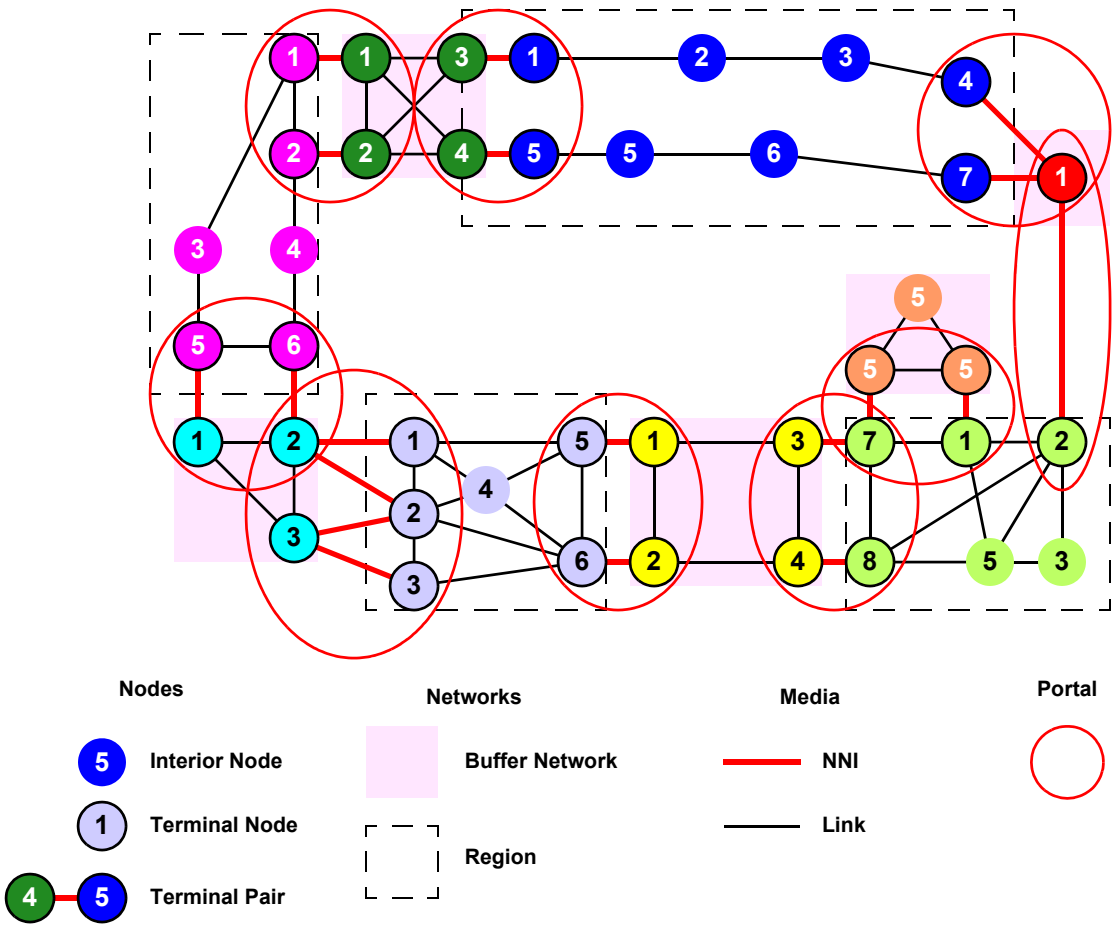


**Figure 1—Definitions and reference diagram for Buffer Network Insertion**

NOTE—A "Link" is never attached to Nodes belonging to different Networks. See "NNI", below.

**2.2 Node**: A switch, bridge, router, or similar device, either logical or physical, connected to other Nodes via any number of Links and NNIs. A Node and all of the Links to which it connects belong one Network. A Node is always either a Terminal Node or an Interior Node, and in addition, may or may not be an Edge Node.

**2.3 Service**: A data service, typically sold by the service provider to a customer, offering data transport among two or more User Network Interfaces (UNIs). Each UNI is a port (not a Portal) of an Edge Node in a Network. A Service is characterized by guarantees by the provider(s) to the customer in terms of connectivity, availability, bandwidth, latency, etc. Only the connectivity aspect will be dealt with in this document. In addition, a Service is characterized by an enumeration of UNIs, Networks, and Portals that provide connectivity among the UNIs. In this document, only "multiregion Services" that span multiple Regions, and thus must traverse Portals, are of interest.

**2.4 Bundle (Bundling):** One or more Services grouped together by configuration, that share the same fate with regard to fault protection. No explicit data encapsulation is implied by a Bundle. A Bundle is a control plane construct whose purpose is to reduce signaling costs.

**2.5 Network-Network Interface (NNI)**: An NNI is similar to a Link in that it carries data between its attached Nodes. It differs from a Link in that it a) is always a point-to-point connection; and b) connects two Terminal Nodes that belong to two different Networks. An NNI is a member of a Portal.

**2.6 User Network Interface (UNI)**: An interface to a customer, attached to an Edge Node.

**2.7 Terminal Node**: A Terminal Node is a Node that belongs to exactly one Network, and is attached to one or more NNIs that (are intended to) connect it to Terminal Nodes in other Networks. It may also be attached to zero or more Links that (are intended to) connect it to other Nodes in its same Network. A Terminal Node is part of one or more Portals, as determined by the NNIs to which it is attached. A Terminal Node may also be an Edge Node.

**2.8 Interior Node**: A Node, belonging to one Network, that is attached to one or more Links that (are intended to) connect it to other Nodes in that same Network, but is not attached to any NNIs, and hence is not a member of a Portal. An Interior Node may also be an Edge Node.

**2.9 Edge Node**: A Node, either an Interior Node or a Terminal Node, that is attached to a UNI to a customer.

**2.10 Portal**: A configured list of one or more NNIs and, by extension from that list, their attached Terminal Nodes. Every Terminal Node in the Portal belongs to one of two Networks, i.e., all of the Portal's NNIs connect the same pair of Networks. A Portal is thus the means by which two Networks can exchange data and/or control messages. At least one of the two Network must be a Buffer Network, and not a Region.

**2.11 Terminal Pair**: The combination of two Terminal Nodes and their connecting NNI. Each of the Terminal Nodes belongs to a different Network. The Terminal Pair is configured to belong to a Portal. The two Terminal Nodes and the connecting NNI may or may not be virtual constructs within a single router, bridge, or switch chassis.

**2.12 Network**: A collection of Nodes and Links that are intended to be fully connected in order to provide Services to customers. A Network of interest to this document includes a) one or more Portals connecting to other Networks; b) zero or more Interior Nodes; and c) the Links that connect its Nodes.

**2.13 Segment**: A Segment is a path through a Network offering data connectivity. It can be as small as a single point-to-point physical Link, or as large as a multi-Network multipoint tree with hundreds of endpoints. It can be all or part of a single Service, or carry nested Services or Segments. The connectivity (or other QoS guarantee) of a Segment can be measured periodically by means of Operations, Administration, and Maintenance (OAM) functions such as those defined in [3], [4], or [5]. A Segment is often (but not necessarily) associated with an explicit data encapsulation that encloses its component Services or Segments.

**2.14 Region:** A Network whose physical topology, configuration, and protection protocols enable it to guarantee, even in the face of some number of Node or Link failures, to provide connectivity (and other features such as Quality of Service or latency) among all of its UNIs and Portals. A Region controls to which the NNIs in a given Portal a packet is to be delivered for transfer to or from an adjacent Network.

NOTE—Do not confuse the Regions defined in this document with the Regions defined for the Multiple Spanning Tree Protocol defined in [2]. There are only so many synonyms to choose from.

**2.15 Buffer Network**: A Network whose physical topology, configuration, and protection protocols enable it to guarantee, even in the face of some number of Node or Link failures, to a) provide connectivity (and other features such as Quality of Service or latency) among all of its NNIs; and b) not propagate failure recovery actions to adjacent Networks.

## 3. Problem statement

There are any number of methods for ensuring connectivity of Services across a Network in spite of failures of Nodes or Links within that Network, including routing protocols, bridging protocols, and protection switching based on continuity check protocols. Any of the above methods will work with the Buffer Network Insertion technique presented in this document, subject to the requirements developed herein. Because protection switching is common in this arena and has a very good worst-case fault recovery time, most of the examples used in this document assume that protection switching is the method is used by all providers' Networks.

Problems with NNIs are difficult to diagnose and correct, simply because the responsibility for diagnosis and maintenance is split between two (or more) organizations. In the interest of simplifying the NNI, sometimes at the

expense of other kinds of optimization, this document makes certain assumptions, and does not explore all alternatives:

a) **Customer address independence:** At an NNI, we do not consider maintaining tables of customer addresses (e.g., the MAC or IP addresses of customer equipment). To do so complicates the NNI, requires extra control messages to govern address distribution and/or learning, and requires the NNI to participate in customer-level OAM operations.
b) **Connectivity independence:** This document assumes that each Network is solely responsible for providing connectivity among its UNIs and its Portals and/or NNIs. If a series of Node and/or Link failures results in a Network becoming split into disjoint subnetworks, then connectivity for some or all of the Services supplied by that Network can be lost. No provision is made in this document for allowing one Network to provide connectivity between the disjoint parts of another Network.
c) **Service congruency:** Many service providers consider that both confining a given Service to a minimum number of NNIs and Links, and ensuring that the path between any given UNIs A and B is the same in both directions, are important characteristics. Other providers consider that scattering a single Service's data over multiple links can have the advantage of more efficient load sharing and/or more optimal routing. Without taking sides as to how a Region should be run, this document opts for simplified maintenance, confining a Service to a single Link, insofar as this is possible.

Other requirements on Networks interconnected by NNIs are addressed in more detail by this document:

d) **Bundling reconciliation:** A Network handles thousands or even millions of Services. Protecting every Service with periodic connectivity tests at high speed (~ hundreds of tests per second) can place excessive burdens on the provider's Links and Nodes. Therefore, Services are grouped together in Bundles for all or part of their path through a Segment so that a single per-Segment periodic test can protect a large number of Services, and failover switching can be signaled efficiently. But, if, for example, one Network groups 4000 Services as {1–2000, 2001–4000} and the other Network as {even-numbered, odd-numbered}, then reconciling their choices which Services take which path becomes difficult.
e) **Fault recovery reconciliation:** Different providers can use different fault recovery methods (GMPLS, MSTP, etc.). These protocols make different provisions (often, none) for interacting with other Networks through NNIs that must be either reconciled or invented.
f) **Failure propagation:** It is essential that a failure recovery action, that is, switching a Service to another path through one provider's Network, not require the neighboring Network to also have to perform a failure recovery action; this could result in a world-wide chain of recovery actions.
g) **Failure set size:** A Network's physical topology and fault recovery method is often designed to provide connectivity, perhaps with a brief pause to perform fault recovery actions, in the face of no more than a certain number (say, *N*) failures of the Nodes and Links comprising a "failure set". For a typical Network, the failure set consists of all of the Nodes and Links in the Network, except for inevitable single points of failure, for example an Edge Node with a UNI to a singly-attached customer. It is undesirable if the construction of a multi-Network Service results in increasing the size of the failure set for that Service above the maximum value over all Networks' failure sets.
h) **Signaling volume:** It is a given that at least those Segments that are physical Links must be protected with high-speed exchanges of continuity check messages in order to detect faults in the 10 ms timescale. Such protection can be built into hardware. When a Segment fails, then either a) each of the Segments or Services encompassed by that Segment needs to be notified quickly of that failure; or b) each of those Segments or Services must be running its own high-speed connectivity tests to detect the failure. The former requires that every Node be capable of quickly generating huge numbers of failure notification messages (e.g. the AIS message of [4]), and the latter requires huge volumes of overlapping connectivity tests that stress the bandwidth of Links near the core of the Network.
i) **Geographical proximity:** When redundant Links or Nodes are provided, it is important to separate them geographically, so that a single adverse event is unlikely to affect them all.

## 4. Existing Solutions

There are any number of schemes for utilizing NNIs, mostly based on particular transport technologies. They are described in the following sections, and include:

a) Hierarchical protection switching (4.1); and
b) Virtual switches (4.2).

## 4.1 Hierarchical protection switching

### 4.1.1 Basic protection switching

A Service of interest to this document is a multi-Network Service, and thus includes at least one NNI in the set of UNIs and NNIs to which a Network provides connectivity. In order to ensure against Node or Link failures, a Network commonly provides at least two Segments for carrying a Service (or a Bundle of Services) between the Edge Node with the UNI and the Terminal Node with the NNI, or Terminal Node and Terminal Node, an active Segment and one or more standby Segments. The continuity of each of these parallel Segments is tested continuously at high speed (~ 10 ms failure detection time) so that if the active Segment fails, the Nodes can failover to a standby Segment. Thus, excepting the end Nodes themselves, the failure set for the Service is all of the Nodes and Links involved in carrying the two (or more) Segments, and the number of failures tolerated is equal to the number of parallel Segments. The parallel Segments together, within one Network, can be called a Network Segment.

The path of the Service from UNI to UNI (to UNI … for a multipoint Service) traverses multiple Networks interconnected by NNIs. Let us assume for a moment that the Service is defined by a single configured point-to-point path (or a single tree) that connects all of the UNIs through the Networks and their interconnecting NNIs. A failure event in a Segment within a Network is confined to that Network by the parallel Segments described in the previous paragraph, and does not affect any other Network. Therefore, the failure propagation problem (section 3, item f), has been solved. However, each NNI along that path is a single point of failure for the path, so the Service as a whole is poorly protected.

One way around this problem (See 4.2 for another.) is to arrange two or more complete UNI-to-UNI paths through the Networks and NNIs, and active and one or more standbys. Each of these paths is a Segment consisting of a serial chain (or tree, for multipoint Services) Network Segments. By running fast continuity tests on each whole UNI-to-UNI Segment, the active and standby Segments can be swapped if needed, and the whole Service protected.

Figure 2 shows an example of a single point-to-point Service connecting UNI L and UNI R using this technique. There are two pairs of Segments in each of the three Networks, each pair colored blue (the primary) and cyan (the standbys). Each pair is a Network Segment, but the Network Segment are not shown explicitly in Figure 2. These two pairs in each of the left and right Networks ensure that each UNI has two paths to each of the two NNIs. In the center Network, the two pairs guarantee two different Services, one between the upper two NNIs, and one between the lower two NNIs. There are two Segments one pink and one orange, running UNI-to-UNI. One passes through the upper two NNIs only, and the other uses only the lower two NNIs. Within each Network, each UNI-to-UNI Segment follows whichever of the half of one pair is currently active. Note that, without the Segment pairs, every Node in all three Networks is in the failure set for this Service. That is, if the cyan+orange and cyan+red paths were not present, then a failure of Node o would kill the orange Segment, and a failure of Node c would kill the pink Segment, and the Service would be lost. The six Segment pairs take Nodes b, c, h, i, o, and p out of the failure set. That is, with them, Node b could take over for a failed Node c and Node p for a failed Node o, so that the orange Segment would follow the path a–b–d–f–h–j–m–**p**–q. ("**p**" is bold, because it is the cyan substitute for the failed m–o–q Segment.) However, if Node a or Node q fail, or the right two of Nodes d, e, f, g, j, k, m, or n fail (e.g. both g and m), both UNI-to-UNI Segments, and thus the Service, can be lost.

The costs of this protection are not small, and include:

a) The failure set size problem (section 3, item g) is present, because for each of the two UNI-to-UNI paths, every NNI and every Terminal Node are in that path's failure set. For a multipoint Service, this set can become extensive, and because each multi-Network UNI-to-UNI Segment is expensive, the number of alternate UNI-to-UNI paths must be kept small, almost invariably only two.
b) The signaling volume problem (section 3, item h) is present, because either every Service (or at least, every Service in a Bundle going to the same set of edge Nodes) must run fast connectivity detection tests, or provision must be made at every NNI to provide a stream of error notifications (e.g. AIS [4]) for up to thousands of UNI-to-UNI Segments.

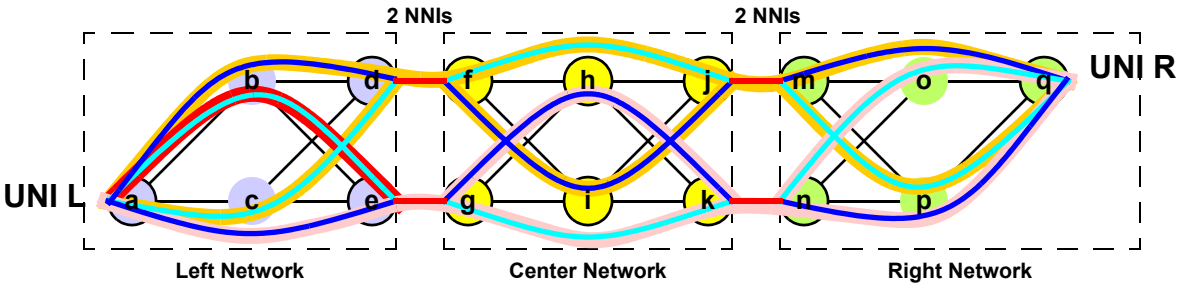


**Figure 2—Hierarchical protection switching**

The result is that the providers are faced with a difficult choice, either to accept the bandwidth burden of a very large number of very fast Network Segment and UNI-to-UNI Segment continuity tests, to accept slower failover times for the UNI-to-UNI Segments, or to try and make the single points of failure (Terminal Nodes and NNIs) ultra reliable (section 4.2).

### 4.1.2 Multipoint issues

Multipoint Services, those with more than two UNIs on a Service, present a special problem. Suppose one configures two multi-UNI end-to-end Segments. Standard means are available in [3] and [4] so that every Edge Node in a Segment can be informed in tens of milliseconds if connectivity between any pair of Edge Nodes in the Segment fails. If only because of the expense of providing redundant, reliable Edge Nodes, the customer typically expects to be able to operate with as many of the UNIs as possible, even if not all are available. Let us suppose a 500-UNI Service is protected by two UNI-to-UNI global Segments. A provider-side failure to reach one UNI results in a rapid failover. But, if the path(s) to more than one UNI fail, (and multiple failures become more likely as the number of UNIs increase) then the provider may have to select the Segment for use with the fewest number of failures. Furthermore, suppose that the paths to UNIs A and B fail in one Segment, and the paths to UNIs C and D fail in the other Segment. Then, the provider must choose between two 498-UNI Segments, even though, if the configuration of the two Segments had been chosen differently, all 500 UNIs might be reachable.

To solve this problem, providers can identify a special Node that is a multicast server, and provide each UNI in the Service with a point-to-point path to that server using hierarchical protection switching. This solves the 498 vs. 500 problem presented in the preceding paragraph, but now the multicast server is a single point of failure for the Service. Adding a second, independent, multicast server as the hub of second UNI-to-UNI Segment returns us to the previous problem, although with much more reliable Links, or the two servers can be made more reliable using a technique similar to that presented in section 4.2. Furthermore, the traffic flow is very heavy in the neighborhood of multicast servers, and the overall bandwidth used across the global concatenation of Networks is usually greater, and never smaller, than if the data is distributed over a tree.

## 4.2 Virtual switches

If Terminal Nodes and NNIs were perfectly reliable, then a single UNI-to-UNI path through Networks and NNIs would be reliable by virtue of the reliability of its component Network Segments. The common means of ensuring this is the combination of a virtual switch with multiple physical media. This solution has also been called “Multi-chassis Link Aggregation” as compared to single-chassis Link Aggregation, defined in [6].

Link Aggregation is a standard technique whereby multiple point-to-point IEEE 802.3 media, all connecting the same two Nodes, appear to the protocol stack above the 802.3 layer to be a single point-to-point Link or NNI. Link Aggregation responds to a failure (or restoration) of any physical medium by shifting traffic among the available physical media. Load sharing among the media is possible.

Link Aggregation makes the NNI between two Terminal Nodes very reliable, but the Terminal Nodes themselves are still single points of failure in a UNI-to-UNI path. Although there is no standard for it as yet, a number of vendors

have developed the idea of a "virtual switch" which, for the present purpose, is a virtual Terminal Node. A virtual Terminal Node is composed of multiple component Nodes, ideally separated geographically, that together appear to all other Nodes to be a single physical Terminal Node, with improved reliability. For reliable connection to the virtual Terminal Node, any Node connecting to the virtual Terminal Node uses Link Aggregation (or some proprietary variant of Link Aggregation), with at least one physical medium connecting to each of the components of the virtual Terminal Node. A failure of one component is perceived by the attached real or virtual Nodes as a Link failure, to be handled by normal Link Aggregation procedures; the other (or the remaining, if more than one) component(s) of the virtual Terminal Node take over from the failed component. Thus, the virtual Terminal Node is more reliable than a single physical Terminal Node would be.

There is a catch, however—the components of the virtual Terminal Node must be connected together, in order to cooperate in producing the fiction that they are a single Node. If the inter-component connection were to fail, and each component acted independently to assume the duties of the other component, the result can be disastrous for both Networks, including failed or duplicate delivery of customer data packets, or even forwarding loops that can cause the Networks to "melt down". Vendors go to great lengths to avoid this situation using multiple Links for the inter-component connection and/or employing the help of the connected non-virtual Nodes to detect the difference between an component failure and an inter-component connection failure. So, although the likelihood of a split-component failure can be made very low, the consequences of such a failure are very high; the consequences are compounded in the political or business sense by the fact that the failure occurs at the junction of two providers' domains.

On the plus side, because the UNI-to-UNI Segments of the hierarchical protection switching solution are avoided, the virtual switch solution is not subject to the failure set size (section 3, item g), signaling volume (section 3, item h) and multipoint Service (section 4.1.2) problems that afflict the hierarchical protection switching solution.

Figure 3 illustrates the same point-to-point Service as that in Figure 2, but using virtual switches. In this figure, Nodes d and e have been paired to form a virtual switch d1, using an inter-component connection (shown in magenta). Similarly, Nodes f and g make Node f1, j and k j1, and m and n m1. Only three Network Segments (Segment pairs) are needed, one in each Network, shown as blue/cyan pairs. While Nodes a and q are, of course, still single points of failure compared to Figure 2, all of the other Nodes have been removed from the failures set. This is carried to the extreme in the lower diagram in Figure 2, where all the Network Segments have been eliminated. But, the price we pay for this simplification is that we now have four single points of failure, the inter-component connections d–e, f–g, j–k, and m–n.

## 5. Buffer Network Insertion

Both hierarchical protection switching (4.1) and virtual switches (4.2) are regularly employed by providers. For the former, the forced choice between the expense of multiple UNI-to-UNI paths and slow failure recovery times is difficult. For the latter, the chances of disastrous failure, though rare, is worrisome. Multipoint Services multiply these difficulties. Neither solution solves the fault recovery reconciliation problem, (section 3, item e) in that end-to-end protection of the customer's Service must be done either with hierarchical protection switching or by virtual switches, and all providers must make the same choice per Service (or Bundle of Services). There is a third alternative, Buffer Network Insertion.

### 5.1 Routing versus protection switching

Protection switching is accomplished by making a decision independently at an endpoint to use one or the other of multiple available Segments, each of which connects to the other endpoints (all of the endpoints, for a multipoint Service), based on end-to-end (-to-end) frequent and continual testing of the continuity of those Segments. Failure recover time is limited to the time required to detect the end-to-end failure plus the time to make and execute the necessary local decisions. However, as discussed in section 4.1, there are limitations inherent in this method.

"Routing" as a general idea includes such techniques as IP routing via RIP or IS-IS, or 802.1 bridging via MSTP. Instead of limiting each Service to a very few fixed, preconfigured Segments, essentially every Link or Node in the Network is available for use by any Service, so that if a physical path exists between any two points in the Network,

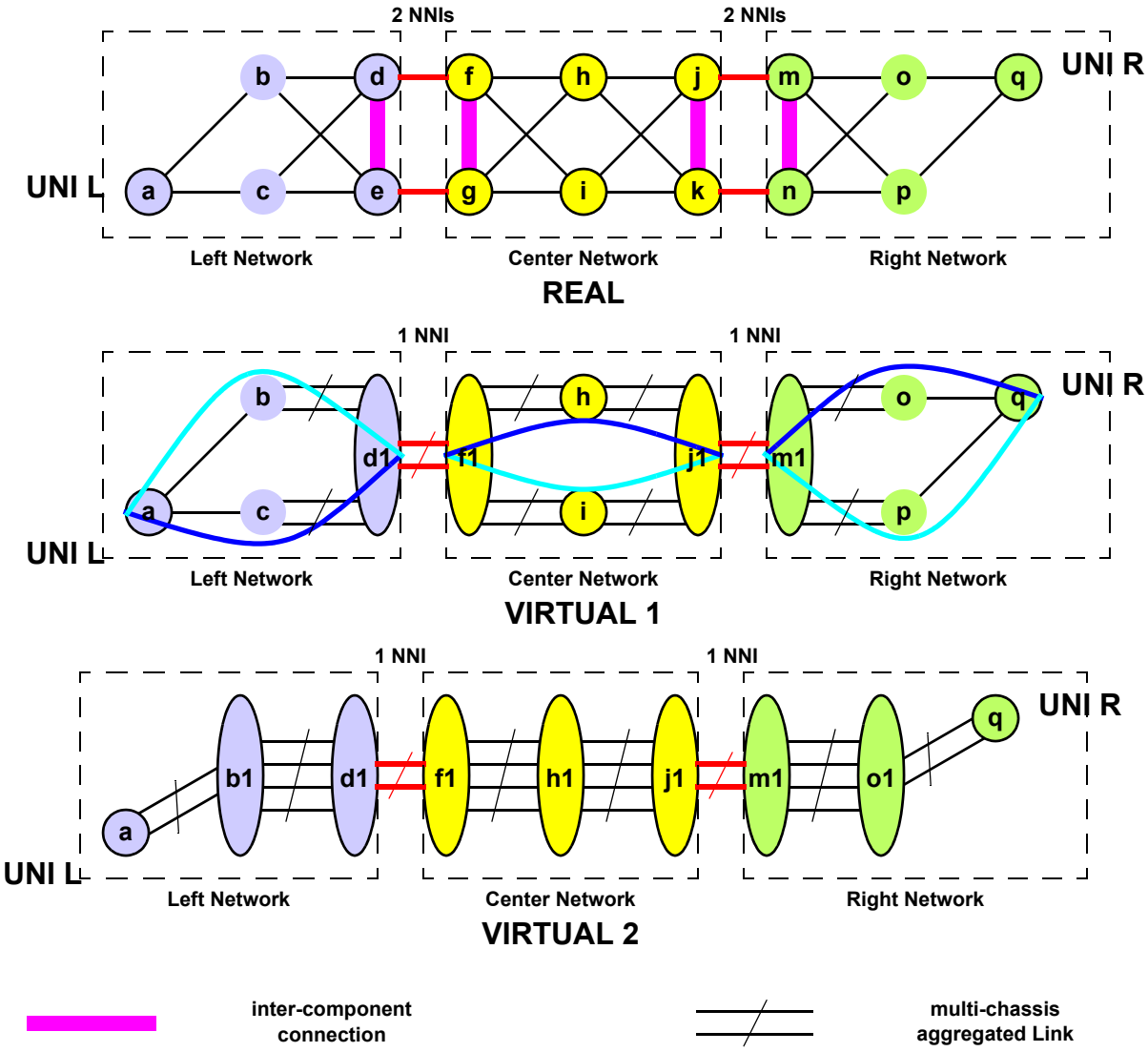


**Figure 3—Virtual switches**

connectivity between those points is assured, regardless of the number or location of failures. However, in order to provide this level of assurance, it is often the case that a failure near one edge of the Network must be signaled via the control protocols from Node to Node across to the other edge of the Network in order to effect a decision to reroute packets to follow a different topology. Thus, the worst case failure recovery time is always much greater for routing than for protection switching.

These poles are apparently irreconcilable; one must accept either:

a) A trade-off between excessive fault detection traffic and alternate path configuration versus an increased likelihood of Service connectivity failure due to an unanticipated combination of faults (hierarchical protection switching, 4.1); or
b) Excessive fault recovery times (routing).

There are examples where both techniques are utilized together, as in [7]. Buffer Network Insertion (section 5) is another example.

### 5.2 Summary of Buffer Network Insertion

In a nutshell, using the definitions in section 2, Buffer Network Insertion has three essential parts:

a) Buffer Network Insertion distinguishes between Portals and their constituent NNIs and Terminal Nodes. The existing solutions discussed in section 4 can be characterized as having only one NNI per Portal.
b) A separate, small Network, the Buffer Network, is inserted between any two provider's Networks, which are then called "Regions". (Typically, this Network is virtual, so no additional physical components are required (see section 5.5).

Both Regions and Buffer Networks are examples of Network, by the definitions in section 2. Furthermore, a Buffer Network is a Region in the sense that it can serve in place of any Region, although this is not always practical. The differences between a Region and a Buffer Network are:

c) A Region guarantees to provide continuity (and QoS, etc.) for each Service, even in the event of the failure of some number of its Nodes and/or Links, among the set of Portals and UNIs configured for that Service.
d) For each Portal belonging to a Region, and each Service configured to traverse that Portal, the Region selects exactly one of the NNIs in that Portal to be used by that Service.
e) A Buffer Network guarantees to provide continuity (and QoS, etc.) for each Service, even in the event of the failure of some number of its Nodes and/or Links, among the set of NNIs selected by the configuration of Portals assigned to that Service and the NNI choices made by the adjoining Regions.

Thus, when responding to a failure, a Region can, if necessary, move a Service from one NNI to another within a Portal. The interposition of a Buffer Network between each pair of Regions solves the problems listed in section 3 as follows:

f) Bundling reconciliation (section 3, item d) is solved by allowing each Region to decide how to apportion Bundles of Services to an NNI within each Portal, and forcing the Buffer Network to re-Bundle Services so that each Service in a Buffer Network Bundle is in the same Bundle in each of the attached Regions This can require as many Bundles in the Buffer Network as the product over all the Buffer Network's Portals of the number of Bundles in the Region attached to that Portal, and even more may be required for load balancing.
g) Fault recovery reconciliation (section 3, item e) is solved by providing a means for Regions to make NNI choices in a manner compatible with their mode of operation (section 5.7) and by providing a set of primitives (section 5.16) for communication between Buffer Networks and Regions.
h) Failure propagation (section 3, item f) is limited in that, although a failure in a Region can cause that Region to alter the NNI assignment within one or more Portals, and thus cause a recovery action in the adjacent Buffer Networks, those Buffer Networks cannot propagate the failure action any further, because they cannot alter NNI assignments.
i) As is true of the virtual switch solution (section 4.2), the elimination of UNI-to-UNI Segments cures the failure set size (section 3, item g), signaling volume (section 3, item h), and multicast (4.1.2) problems.

### 5.3 Buffer Networks versus Regions

To ensure connectivity without UNI-to-UNI Segments requires redundant NNIs between connected Regions. But to prevent the propagation of failure events from one Network to the next requires that the Network experiencing the failure not change NNIs. To to guarantee connectivity among $P$ Portals, each with $N_P$ NNIs, against $N$ failures, the number of required Segments $S$, each with $P$ endpoints, is given by:

$$S = \frac{1}{2}\sum_{P}\left((N+1)N_P\prod_{i\neq P}N_i\right) \qquad (1)$$

Thus, $S$ grows in proportion to ${N_P}^P$ for a Buffer Network. Since a Region only needs to ensure connectivity only to its Portals, a Region requires only $S = (N+1)$ Segments, each with $P$ endpoints. Thus, a protection switched Region can scale to have vastly more Portals than a protection switched Buffer Network. Since the Buffer Network is required to solve the Failure propagation problem and also solves the Bundling reconciliation problem, while allowing the Region to scale up to large numbers of Portals and/or Edge Nodes, this division into two Network types is worthwhile.

Note that there is nothing that prevents a Buffer Network from having Edge Nodes and UNIs, and nothing that prevents one from having more than two Portals, except the burden of meeting the requirements of equation (1).

### 5.4 Protection-switched Buffer Network

Although there are certainly different ways in which a Buffer Network can be realized, one will be described, the Protection-switched Buffer Network. As mentioned in section 5.1, a combination of routing techniques and protection switching techniques is required. However, instead of adding a protection switching mechanism to a base of routing, as in [7], we will add one routing mechanism to a base of protection switching.

In the simplest case of a Buffer Network with two Portals is shown in Figure 4. In order to be able to account for any single Link or Node failure, and in order to allow either of the adjacent Regions to alter NNI choices without affecting the other Region, eight Segments are required as per equation (1): 1–3, 1–2–4–3, 1–3–4, 1–2–4, 2–1–3, 2–4–3, 2–1–3–4, and 2–4. This set provides an active Segment (cyan) and a standby Segment (blue) between each pair of Terminal Nodes residing in different Portals, 1–3, 1–4, 2–3, and 2–4. This is the protection switching base of the Protection-switched Buffer Network.

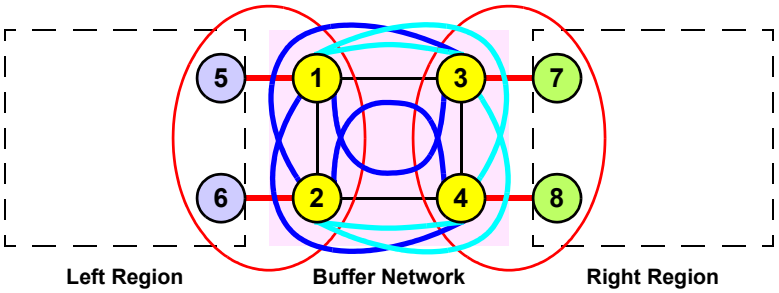


**Figure 4—Two-Portal dual NNI Buffer Network**

NOTE—Although only one NNI is shown attached to each Terminal Node, the set of required Segments depend on the enumeration of connections to NNIs, not the list of Terminal Nodes.

The routing mechanism added is the ability of a Terminal Node in the Buffer Network to be told by the adjacent Region that the NNI connecting them is now the preferred NNI for a Bundle of Services, and for that information to be relayed to the other Portal(s), so that all involved Terminal Nodes can change Segments. For example, suppose a Bundle is using the 4–8 NNI in Figure 4, and for reasons not apparent to any Node in the Buffer Network, the right Region elects to move that Bundle to the 3–7 NNI. Whether the right Region notifies Terminal Node 3 that its NNI is the new choice, or whether it notifies Terminal Node 4 of that fact, or both notifications take place, Node 3 or Node 4 (or both) must somehow inform Node 1 and Node 2 of that change. Each Bundle moved in the right Region can affect a number of Bundles in the Buffer Network.

There are many ways for the Buffer Network's Terminal Nodes to notify each other of NNI shifts:

a) A Terminal Node can multicast a notification in the data plane to all Terminal Nodes in the Buffer Network through one or more control paths reserved for that purpose, and repeat it to ensure delivery.
b) Each Terminal Node could flood frequent updates, either in the data plane or hop-by-hop through the Terminal Nodes, giving its opinion of all Bundle–NNI bindings.
c) Some combination of techniques, to ensure both long-term synchrony of the NNI–Bundle information, and provide fast fault reaction, such as hop-by-hop updates, along with data plane multicasts of events.
d) The connectivity check messages (CCMs) used to diagnose the connectivity of the Buffer Network's Segments could carry information about NNI usage.

Assuming that the Buffer Network uses the CCMs of [3] and [4] to maintain the integrity of its Segments, item (d) seems the obvious choice.

If a Terminal Node loses all contact with the rest of its Network, and is not also an Edge Node, it would be appropriate to indicate a failure of its attached NNIs, so that customer data packets will not be unnecessarily discarded.

## 5.5 Virtual Terminal Pairs

Assuming that a common protocol for Buffer Networks can be developed, then it is clearly advantageous to combine each Terminal Pair, e.g., 3–7 or 6–2 in Figure 5, into a single chassis, virtualizing the Terminal Nodes and their shared NNI. (This is the plan described in [1].) The advantages and disadvantages of Virtual Terminal Pairs include:

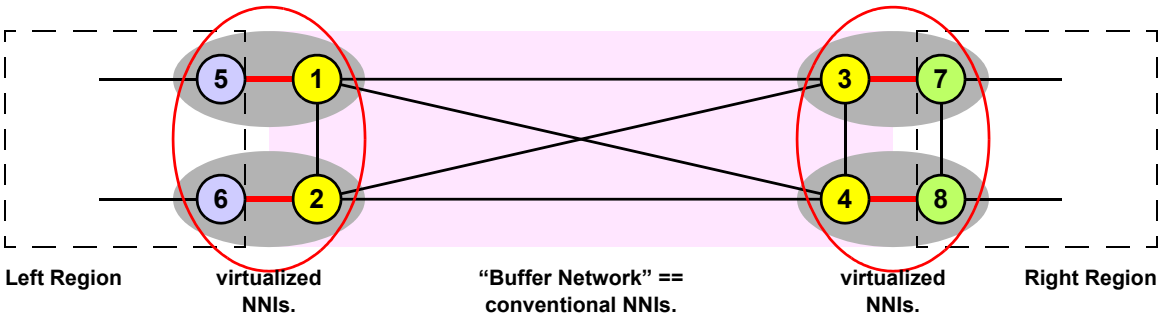


**Figure 5—Virtual Terminal Pairs**

a) The Buffer Network's inter-Portal Links, now correspond to most people's notion of an NNI.
b) Each provider's Terminal Nodes operate a standard protocol, irrespective of the protocol choices made inside the providers' Regions.
c) The total number of chassis is reduced.
d) The Bundling requirements for the Buffer Network are complex (section 5.6), and must be coordinated between the providers.

Notice also that there can be a Link belonging to the right Region between Nodes 7–8, as well as the previously-discussed Buffer Network Link 3–4. If Terminal Pairs 3–7 and 4–8 are each combined into two chassis, then the separate identity of Links 3–4 and 7–8 must be preserved, even if creating the virtual Terminal Pair caused the two Links to share the same physical medium. For example, the right Region, due to some fault, or simply to achieve proper load balancing, might have to deliver one or more odd-numbered Services to Terminal Node 9 via Link 7–8, and due to a failure of Link 2–3, the Buffer Network might have to employ Link 3–4 for all odd-numbered Services. In that case, it would be essential to differentiate, for example, between packets on an odd-numbered Service passing from Node 4 to Node 3 on Link 3–4, on their way from right to left, from those passing from Node 8 to Node 7 on Link 7–8, on their way from left to right.

## 5.6 Bundling

A simple version of the Bundling problem (see [9] for another angle on this and other NNI issues) is illustrated in Figure 6. This figure assumes that the Service identifiers are the same in all three Networks, so that only the question of which Services share the same Bundle is relevant.

Let us suppose that the repertoire of Segments for this Buffer Network are 1–3, 1–2–3, 1–4, 1–2–4, 2–3, 2–4–3, 2–4, and 2–3–4. Other choices are possible; this topology is slightly richer than necessary. Also, additional Segments could be provided (e.g., 1–3–4) to allow for some, but not all, combinations of two Link or Node failures.

We notice first that the Buffer Network requires four Bundles, whereas each of the attached Regions have only two each. This is necessary so that a data packet received on the NNI selected by one of the Regions can enter the other Region on the correct NNI, as selected by that other Region. Depending on other factors, the Buffer Network may choose to subdivide the Services into even more Bundles. For example, if the 2–4 Link were omitted from the diagram, the Buffer Network would still be able to meet all of its guarantees, but it might want to divide the Services

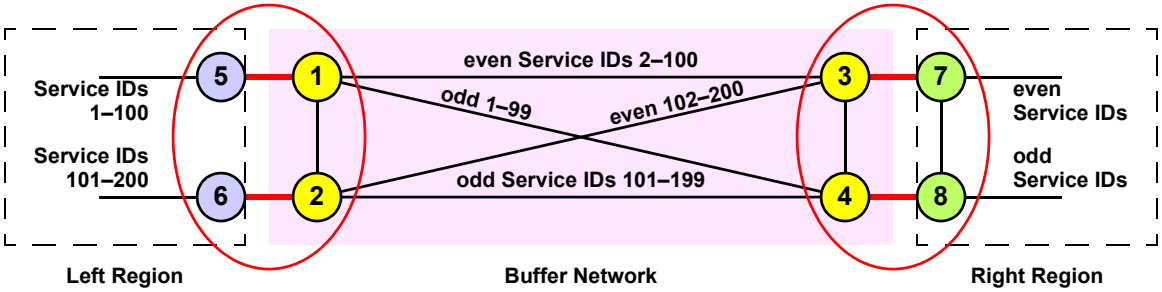


**Figure 6—Bundling issue**

into three groups with more-or-less equal bandwidth requirements. To achieve both even load balancing over the three Links, while accommodating NNI reassignments by the Regions, it might want to define 12 Bundles.

NOTE—In Figure 6, if only Links 1–2 and 3–4 were present, and none of Links 1–2, 1–4, 2–3, or 3–4, this would not be a satisfactory topology for a Buffer Network. For example, if that very limited topology, if the right Network moved all Services to the 3–7 Link, and the 1–2 Link failed, there would be no way to get the even 102–200 Services from Node 3 to Node 6, and the Buffer Network would have to either propagate the fault recovery action by forcing the left Network's Services 101–200 to the 5–1 Link, or lose connectivity for those Services.

## 5.7 Making the NNI choice

Buffer Network Insertion requires the Region to determine, for each Service on each Portal, which NNI is to carry that Service across that Portal, and somehow conveying that choice to some number of the Terminal Nodes in the Portal. This is not necessarily a trivial task, and must be explored in the context of the various technologies available for effecting fault recovery actions in a Region.

The fact that Buffer Network Insertion can work with many different styles of fault recovery protocols in the Regions does *not* necessarily mean that those fault recovery protocols can work without alteration when connected via Buffer Networks. The problem is that, while most all fault recovery protocols are naturally able to make a choice among the Terminal Nodes in each Portal, they are not necessarily equipped to notify the Terminal Nodes of that choice. Hence, the following subsections analyze several different protocols for Region fault recovery:

a) Active versus passive notification is discussed (section 5.7.1);
b) Multiple Spanning Tree Protocol (section 5.7.2);
c) Hierarchical protection switching (section 5.7.3);
d) Multiple parallel subnetworks (section 5.7.4);
e) Multi-Chassis Link Aggregation (section 5.7.5).

More need to be examined, of course, MPLS/VPLS and Shortest Path Bridging, in particular.

### 5.7.1 Active versus passive notification

For most possibilities for Region protection, there are various choices for how a Region notifies a Buffer Network of its NNI choices:

a) Ideally, the Region notifies all of the Terminal Nodes in a Portal, at very nearly the same time, of all choices and changes in choices.
b) When a Region changes its idea of which NNI is to be active, it can notify only the Terminal Node attached to the newly-activated NNI.
c) When a Region changes its idea of which NNI is to be active, it can notify some Terminal Node in that Portal, but not necessarily the one attached to the newly-activated NNI.
d) When a Region changes its idea of which NNI is to be active, it can notify only the Terminal Node attached to the no-longer active NNI, or can notify some Terminal Node of the de-activation.

e) The Region can make its choice quietly, redirecting the data streams, but making no explicit signal.
f) Pre-configured priority lists of Services vs. NNI choices can be used to back up any of the above methods.
g) The Terminal Nodes in a Portal can exchanged signals among themselves in response to any of the above events, in order to communicate the choices made by the attached Regions.

As discussed in the following subsections, some candidates for Region fault protection protocols naturally supply signals to the Terminal Nodes, and some do not. We can imagine adding such signalling capabilities to some protocols. In general, item (e) is the least reliable, because it can be difficult to coordinate all of the paths that different Services take to make a clean switch-over. Some old traffic is likely to take the old path for a short time after the first new packet takes the new path, and this can cause flapping of the choice.

Because some time is required for a Region to make the NNI choice, and some delay in propagating that choice to the Buffer Network Terminal Nodes attached to an adjacent Region, there is inevitably a window during which the Buffer Network delivers a Service's packets over the "wrong" NNI, as perceived by the receiving Terminal Node. Customer address independence (section 3, item a) is thus violated, at least briefly; multiple faults in the Buffer Network can extend this period. It is up to the receiving Region whether to forward packets delivered over the wrong NNI to their intended destinations, or whether to discard them.

### 5.7.2 Multiple Spanning Tree Protocol

If a Region uses the MSTP protocol of [2] for fault recovery, then the Layer 2 Gateway Protocol (L2GP) defined in [8] can be configured to make a consistent decision. L2GP prevents even temporary forwarding loops, and it will enable both NNIs 3–5 and 4–6 only if the Region becomes divided (by multiple failures). (A divided Network is a separate issue; see section 5.10.) Thus, there are already standards for MSTP Regions to support Buffer Network Insertion. However, L2GP works only for a Region with a single Portal.

NOTE—Do not confuse the Regions defined in this document with the Regions defined for the Multiple Spanning Tree Protocol defined in [2]. There are only so many synonyms to choose from.

### 5.7.3 Hierarchical protection switching

Figure 7 is a copy of Figure 2, with the left and right Networks now Regions, and the center Network now a Buffer Network. A fundamental issue with end-to-end hierarchical protection switching can be discerned from this figure: The rest of the global network must supply the left Region and right Region with two separate complete services, one to link Node d to Node m, and one to link Node e to Node n. In other words, if both Nodes m and e fail, then the Service fails, even though it might be possible to construct a path from Node d to Node n. The reason is that the end-to-end Segments must be kept separate. If, for example, the path a–b–d–f–k–n–p–q were formed, the CCMs of [3] and [4] would detect a cross-connect failure.

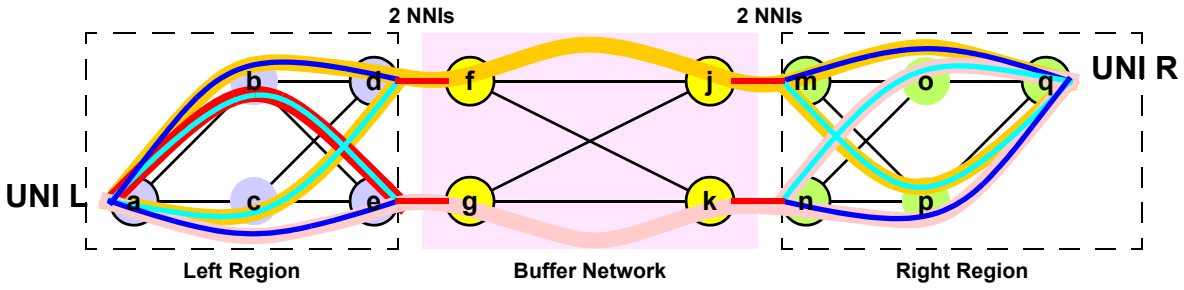


**Figure 7—Hierarchical protection switching and Buffer Network Insertion**

Note that a Buffer Network as described in section 5.4 could be inserted into Figure 7 to support the two hierarchical protection switched Regions, even though it is capable of cross-connecting the two end-to-end Segments. This is because the right and left Regions both make the switch to the alternate end-to-end Segments at the same time. The Buffer Network responds to both choices, and supplies the required connectivity. (Although there might be a brief period in which cross-connect errors could be reported.)

Assuming that the Buffer Network in Figure 7 is configured to supply only Segments that connect Node d to Node m and connect Node e to Node n, no signaling of preferences by the Regions is required to make the network operate correctly. The Buffer Network does its best to supply the two required Services, and the fault protection is less than ideal. (But, see section 5.7.4 for an improvement.)

The protection of Figure 7 could be improved. The Buffer Network could supply two additional Segments f–k and g–j, and two additional end-to-end paths could be created to use them. This, of course, further increases the signaling load on the networks, and as more networks are added to the picture, the number of paths grows exponentially.

### 5.7.4 Multiple parallel subnetworks

In a Portal with more than one NNI, the NNIs typically reside in different Terminal Nodes. Since they do not reside in the same Node, simple protection switching cannot make the decision as to which to use. A protection switched Region can also use Multiple parallel subnetworks, as described in this section, to make the NNI choices.

In a Region, on can Bundle Services together on the basis that each Service that is configured to pass through the same set of Portals and Edge Nodes is placed in the same Bundle. To reduce the number of Bundles at the expense of forcing some Nodes to incur unnecessary fault recovery actions when a failure affects connectivity to some other Node, Services with different Node sets can be Bundled together, up to the limit of one Bundle for the whole Region.

For each such Bundle, $N_B$ Segments are created. Each Segment has an endpoint in each Edge Node in the Bundle[1], and has one endpoint in some NNI in every Portal in the Bundle.[2] $N_B$ is typically equal to $N_P$ in equation (1), but since a Service, and therefore a Segment, can touch more than one Portal, and since different Portals can have different numbers of NNIs, $N_B$ is chosen to balance the needs of the Network; the administrator may have to terminate multiple Segments in one Terminal Node, or not include some NNI(s) of a Portal in any of the Segments of some Bundle, if $N_B$ is greater than or less than, respectively, of some Portal's $N_P$. The reason for calling this the "Multiple parallel subnetworks" technique is that, since the two NNIs on the same Portal are not in the same Node, there can be no Network Segment encompassing the pair. Operation of the connectivity check protocols in [3] or [4] ensure that both the Terminal Nodes and the Edge Nodes in the Segment have the same picture of the Edge Node-NNI connectivity within the time constraints of these protocols, so all will know whether or note all Segments have connectivity. The Terminal Nodes and Edge Nodes can therefore select one or the other of the Segments (subnetworks) to be the active Segment and the others to be standby Segments. All will chose the one Segment that has complete connectivity, or make a pre-configured choice if both are complete. Since all of the Terminal Nodes participate in determining the health of the Segments, all know the resultant choices.

Figure 8 shows an example of these Multiple parallel subnetworks. The Terminal Nodes a–f, and Interior Nodes 1–6 (not Edge Nodes) and Interior Nodes (and Edge Nodes) u–z, as well as the Links belonging to the Region have been colored in three different ways: The Interior Nodes and Terminal Nodes (all non-Edge Nodes) are colored either orange (a–c and 1–3) or blue (e–f and 4–6) as belonging to either the "orange ring Segment" or the "blue ring Segment," and the Interior Nodes u–z, that are also Edge Nodes, are colored yellow.

Note that the orange and blue ring Segments are independent, but that each Edge Node connects to both the orange and blue Segments. We will assume for sake of this explanation that there is only one Bundle of Services configured, with two Segments, orange and blue. However, in a typical Region, there would likely be several orange and several blue Bundles, corresponding to different subsets of Portals and Edge Nodes required by different Services.

For any given Bundle, every Node in each of its Segments can be made aware, via the CCMs of [3] or [4], whether or not all of the Nodes (or even NNIs and UNIs) in the Segment are able to communicate with each other. For example, each of the Terminal Nodes b and e in the uppermost Portal knows whether its ring Segment (orange for Terminal Node b, blue for Terminal Node e) is available to carry the data for its Bundle.

In Figure 8, then the choice for the Bundle of Services is a choice between the orange and blue rings. As outlined in section 5.7.1, the choices can be made, and the adjacent Buffer Networks notified, relatively easily:

1. For [3] or [4], this could be done with an Up MEP in a management port.
2. For [3] or [4], this could be done with an Up MEP at the NNI port.

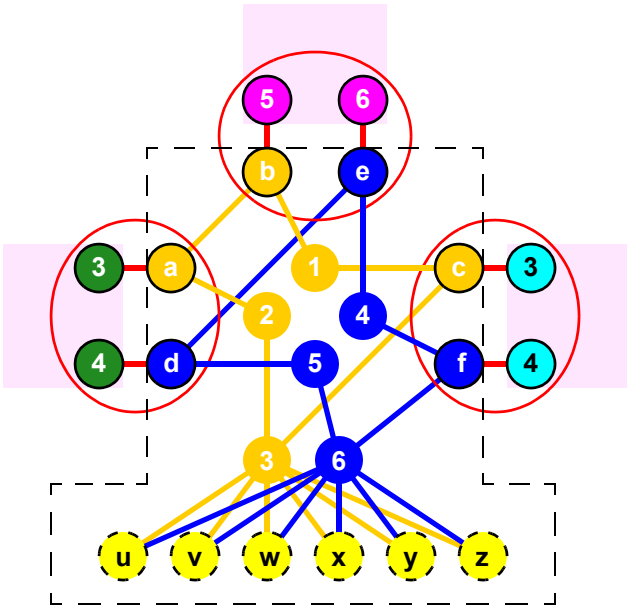


**Figure 8—Multiple parallel subnetworks**

a) Every Edge Node must make a choice of which ring Segment to use for data coming into the Region from the UNIs, even if all data received from either the orange or blue Segments is passed on to the UNIs for output. Since all of the UNIs use the same algorithm—use the Segment that is complete, or use the pre-configured first choice if both are complete—then only one Segment will be chosen.

b) Every Terminal Node knows, for every Bundle it serves, whether all of the other Nodes in the Segment of that Bundle served by that Terminal Node are in communication or not. It can therefore report across the NNI whether the Bundle is able to use that NNI or not.

c) The Terminal Nodes of the Buffer Network can exchange the information received from their partners across the NNIs to decide which NNIs are in use. If a Bundle has no good Segment, neither NNI is used, if one, that NNI is used, and if more than one good Segment, the one with the best pre-configured priority [the same as the choice made in item (a), above] if used.

If no Node or Link fails in the Region, connectivity is guaranteed first by ring protection, and then by priority choices. If one Node or Link fails, then that ring repairs itself. If two failures occur, the worst case is that one of the orange or blue ring Segments is partitioned, in which case the other ring Segment will take the load. If one NNI fails, then the affected Segment, either orange or blue, is taken down and all of the Portals must make a switch. If multiple failures occur, affecting more than one Segment, then the safest course is to shut down all of the Segments and lose connectivity. If additional information is carried in the CCMs (see sections 5.9, 5.10 and 5.16), then some ability to work safely in the presence of multiple faults can be maintained.

### 5.7.5 Multi-Chassis Link Aggregation

It is possible to make the Terminal Nodes of a Portal appear to the other Nodes in the Region to be a single virtual Node, and use an extension of Link Aggregation Control Protocol (LACP). Normally these protocols allow two Nodes that have multiple Links connecting them to treat the multiple Links as a single Link. Normally, load sharing is used to take advantage of the added bandwidth. Providers prefer to base the load sharing on Service identifier. “Multi-Chassis Link Aggregation” differs from this in that at least one end of the Bundle of Links terminate on multiple Nodes.

In the general case, this requires that the components of the virtual switch share learned MAC address information. For a number of protocols, either their state machines’ state must be shared across the connection between them in order to look like a single device to the external Nodes, or one component must relay all such frames to and from the

other component for processing. Protocols include LACP, spanning tree protocols, VLAN pruning protocols, routing protocols, and much more.

This requirement for exchanged information can be simplified considerably if the load sharing is by Service, rather than by some other criterion such as the IP 5-tuple, because then learned MAC addresses need not be exchanged. The Link between the components of a virtual switch can incorporate special hardware, and uses special frame formats, so that the components of the Virtual Switch can trade full backplane header information along with each frame. Eliminating the MAC address learning requirement could eliminate the need for the Link between the components to carry backplane information, and makes standardization practical.

In terms of this document, the entire Buffer Network is, in a sense, equivalent to the VSI. The data Segments in the Buffer Network are presumably tagged (S-VLAN, I-SID, MPLS+Pseudowire, etc.), and these tags replace the VSI backplane headers. Some means of interchanging the shared protocol state is also required, but these protocols can be carried over similar data Segments. So, *if* one uses some sort of multi-chassis Link Aggregation to connect the Interior Nodes in each Region to the Terminal Nodes of a Portal, *then* the entire collection of Terminal Pairs and the Buffer Network appear in many respects to be a single Very Large Virtual Switch. On the other hand, if some other technique of selecting Terminal Nodes is used, e.g. Multiple parallel subnetworks (section 5.7.4) or left-right subnetworks (Section 4.1.2), the difference between Buffer Network Insertion and giant virtual switches becomes more clear.

This document is not the place to go too deeply into Multi-Chassis Link Aggregation. We will simply observe that the requirement to, for example, connect a Region based on Multi-Chassis Link Aggregation to a Region using some other scheme, may impose some requirements on Multi-Chassis Link Aggregation.

## 5.8 *Not* making the NNI choice

Some fault recovery protocols for Regions are perfectly comfortable receiving data for a single service on any NNI, or even multiple NNIs, and some administrations of Regions do not care about Service congruency [item (c) in 3], or consider efficient load sharing more important.

There are Regions, such as the lower left Region in Figure 1, that are well-connected internally, so that it does not matter a great deal on which NNI a given Service is received (although the network may certainly have a preference, because of path length considerations). In other cases, such as the one in Figure 8, the upper right Region in Figure 1, or the former two Regions when certain faults have occurred, connectivity can be provided only if the Buffer Network delivers the data over the preferred NNI.

It is therefore reasonable to allow the NNI choices made by a Region to include the “no preference” option. This allows the other Terminal Nodes in the Buffer Network to make a convenient choice as to which NNI(s) a given Bundling is delivered. However, this decision is made at the convenience of the transmitting Terminal Node and in the absence of customer address information (section 3, item a).

It is also reasonable, if none of the administrators of the Regions connected by a Buffer Network object, to allow a Region to choose to send data for a given Service or Bundle on any NNI, or multiple NNIs, whether or not one is preferred for receiving data.

NOTE—If neither Region connected to a Buffer Network has a preference for the incoming and outgoing NNI for any Service or Bundle, then Buffer Network Insertion is largely equivalent to some multi-chassis LACP solutions. However, the loss of a Node or Link can necessitate expressing and respecting NNI preferences. Similarly, interoperation with non-mLACP Regions and/or respecting another Region’s administration’s requirement for Service congruency (section 3, item c) requires mLACP to support NNI preferences for outbound data.

## 5.9 CCM enhancements

In section 5.7.4, “Multiple parallel subnetworks”, we mentioned that the current CCM messages as defined in [3] and [4] can be made more robust against multiple failures. The reason is that we have to be sure that all of the Nodes in a

Segment agree on the NNI selection for their shared Segment when choosing among Segments, all of which are at least partially failed.

The details of what features are needed are not settled at this time. The primary candidates are:

a) Adding a "Segment choice" TLV to the Continuity Check Message (CCM) of [3] or [4], or some corresponding message for another OAM technology. This allows every MEP to know the choice made by every other MEP.
b) Adding a per-remote-MEP RDI indication supplies enough information that each MEP can determine exactly the subset of MEPs in the Segment that have two-way connectivity, and no connectivity at all to any other Nodes. This allows MEPs to choose the best among the partially-failed Segments, while ensuring against forwarding loops caused by one-way connectivity.

### 5.10 Divided Networks

If a Network has more than one Terminal Node in at least one Portal, then no matter what protocol(s) are used to route traffic within that Network, there is at least one combination of Link failures that results in the Network being divided into two or more pieces, each connected to an NNI, as shown in the top center (blue) Region in Figure 1. In this particular example, there are two independent chains of Nodes and Links, each providing Portal-to-Portal connectivity, and neither one connected to the other.

By definition, the two parts of the Network are disconnected, so neither one knows for sure whether or not the other one is providing connectivity. Whether this is a feature, providing connectivity when it would otherwise be lost, or a bug, causing endless forwarding loops that can melt down the global network, depends on how the neighboring Networks handle the situation.

a) If exactly one neighboring Network enables multiple NNI to pass data to and from the disconnected parts of the divided Network (e.g., the upper right single-Node red Buffer Network in Figure 1), then connectivity can be maintained even in the face of some combination of multiple failures, specifically, when those multiple failures divide a Network.
b) If more than one neighboring Networks enable multiple NNIs to pass data to and from the disconnected parts of the divided Network (e.g., both of the Buffer Network adjacent to the top center Region in Figure 1), then a forwarding loop results.
c) If two consecutive Networks both split, and each half of each Network maintains at least one NNI with only its corresponding half-Network neighbor, there is no immediate problem other than a possible loss of connectivity, and all of the other issues in this section are merely carried over to the other Networks connected to the pair of split Networks.
d) A Buffer Network with only two Portals does not need to perform any address-based packet switching, e.g., switching by MAC address or IP address; everything that comes in one Portal goes out the other Portal. But, if on one Portal, two NNIs are enabled, the Buffer Network has, in effect, three Portals, and must either forward all packets received on any one of the three to the other two, or must perform address-based packet switching. This is a very big step to take, particularly when the "Buffer Network" is actually virtualized portions of the switches in the facing Regions as in section 5.5 and Figure 5.

In summary, in the absence of any global loop detection or prevention protocol, it seems safest to build each Network to some limit to the number and type of faults it protects against, and to suffer the consequences when that limit is exceeded, rather than to take the chance of inducing forwarding loops that can debilitate the entire global network. That is not to say that providing connectivity to a divided network absolutely must be prohibited; only that this capability should be a secondary consideration, only employed where explicitly configured, and even then not lightly undertaken.

### 5.11 Identities

In order to configure the path of a Service through the global network of Networks, and to ensure unambiguous passage of information about NNI usage between Networks, globally unique identifiers for NNIs would be most

convenient. [3] and [4] provide means for such identification. Similarly, each Network must have a global identity, and as mentioned in section 5.16, each Portal requires an identity that is, together with the Network ID, globally unique, if not globally unique on its own.

An unambiguous means for identifying individual Services in the control messages of section 5.16 is needed. Since the customers' data packets must also be unambiguous as they cross the NNI, the data plane identifiers can be incorporated into the protocols of section 5.16 with a suitable enumeration to select the particular data identifier type (IEEE 802.1ah I-SID, MPLS label, etc.).

To be safe, when a Region's Terminal Node informs the Buffer Network's Terminal Node of an NNI reassignment, it would transmit a list of affected Service identifiers. If the data's identifier space supports only, say, 4k identifiers, then this is practical, because a 4k bit vector requires only 512 bytes. There are, however, 16M IEEE 802.1ah I-SIDs, so a list, even a bit vector, is impractical. In such cases, the Bundling (if any) used by the signaling Region can be conveyed in bulk and periodically verified, and the Region-specific Bundle identifiers used to signal across the NNI.

Similarly, Bundle identifiers and their associated lists of Service identifiers can be incorporated into the "routing" part of the Protection-switched Buffer Network protocol, or they can be pre-configured, with the usual trade-offs between protocol complexity and the difficult of configuring identical information in multiple Nodes.

Of course, if the Buffer Network and its neighboring Region use different transport technologies, they must agree on one technology to use across the NNI, and one or both of the Terminal Node must translate between that technology of the NNI and the one used inside its Network.

## 5.12 Global forwarding loops

If the Regions and Buffer Networks are configured in a closed loop (as shown in Figure 1) and some Service is configured all the way around that loop, then there will be a global forwarding loop. Depending on how and where bandwidth limitations are enforced, this could seriously disrupt a Network, and certainly will disrupt the operation of the looped Service. Of course, providers are well aware of this possibility and attempt to avoid configuring such mistakes. (Whether a global loop-prevention or loop-detection algorithms, operating only on Portals, is possible, is an item for future study.)

Assuming we have no configured loops, there is still the problem that any time that two NNIs in the same Portal are enabled, we have a potential forwarding loop. There are two reasons that this can occur:

a) Multiple NNIs in a Portal may be enabled briefly during an NNI reassignment, while the Nodes of the Buffer Network come to agreement.
b) If a Region becomes divided by Node or Link faults into two or more disjoint parts, then enabling an NNI to each part, presumably through the same Portal, may be the only way to restore connectivity.

We could avoid case (a) by running a more sophisticated protocol either in the Buffer Network, in the Region, or both, to prefer dropping packets, perhaps unnecessarily, rather than ever enabling multiple NNIs. The Layer 2 Gateway Protocol in [8] is an example of a protocol that does this. Or, we could take our chances on these loops, on the grounds that they can occur only briefly.

It is more difficult to deal with case (b). For point-to-point Segments, there can be no bifurcation, and there is no problem. As the number of Nodes in the Segment grows, so does the possibility of bifurcation and the desire for the long-term enabling of multiple NNIs to ensure connectivity grow. Unfortunately, the possibility of bizarre connectivity errors that would cause long-term forwarding loops also grows with the number of Nodes. Further exploration is needed before case (b) can be considered.

## 5.13 Trade-off example

Perhaps the best illustration of the trade-offs being made by Buffer Network Insertion is shown in Figure 9. This is similar to the example shown in Figure 6, but with a critical difference—the Links between Terminal Nodes 1–2 and 3–4 have been eliminated, leaving only the four Links 1–3, 1–4, 2–3, and 2–4. In Figure 9, two Segments, the

primary and the standby Segments for connection Terminal Nodes 1 and 3 are shown. The primary Segment is 1–3, and the standby Segment is 1–4–2–3!

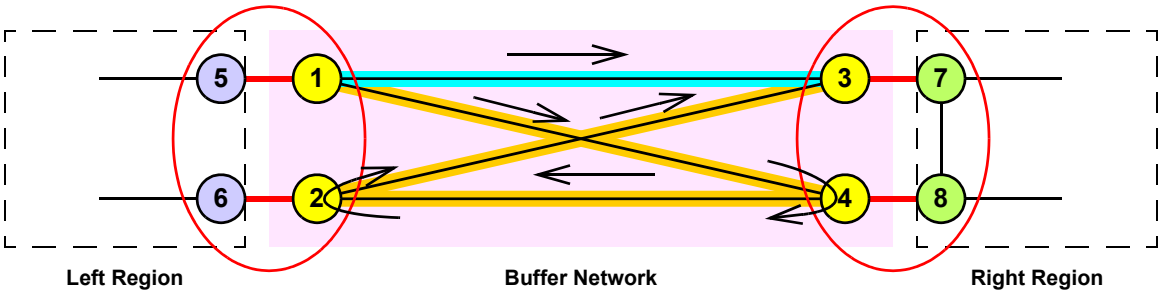


**Figure 9—Illustrative example of Buffer Network**

Why this torturous route? Because it avoids the Failure propagation problem (section 3, item f). Suppose a certain Service is selected by the left Region for NNI 5–1, and by the right Region for NNI 3–7. Then clearly, Link 1–3 is the primary Segment. But, if that Link fails, and the standby Segment 1–4–2–3 is not there, one or both of the left and right Regions have to shift to a different NNI in order to use one of the remaining three Links, which violates our rule that simplifies the life of a Region.

NOTE—This is *not* to say that Figure 9 is a good way to build a Buffer Network! The Buffer Network in Figure 6, with Links 1–2 and 3–4 restored, is the preferable physical topology, in which case the zig-zag Segment becomes a fourth-best standby option, and likely is not even configured.

Taking a slightly different scenario on Figure 9, suppose that another Service has been selected for NNI 6–2 by the left Region and NNI 3–7 by the right Region, so is taking the 6–2–3–7 path. Again, Link 1–3 fails. This Service is not affected by this failure. But, if some failure in the left Region causes that Region to select NNI 5–1 for the Service, Segment is 1–4–2–3 suddenly becomes critical. In its absence, the left Region's failover from NNI 6–2 to NNI 5–1 would have to use Link 1–4, forcing a failover in the right Region. In other words, we have to choose between:

a) Suffering the Failure set size problem (section 3, item g);
b) Suffering the Failure propagation problem (section 3, item f); or
c) Solving both problems at the same time, at the cost of the "zig-zag" standby Segment in Figure 9.

Buffer Network Insertion chooses item (c), on the grounds that the Failure set size and Failure propagation problems are very real, and not trivial to solve, and the solution is well worth the cost.

## 5.14 Only Buffer Networks

In a Buffer Network, the number of pre-configured Segments required to solve the problems in section 3 grows very rapidly as the number of Terminal Nodes and/or Portals increases. This does not necessarily limit a Buffer Network to only two Terminal Nodes per Portal, or even to only two Portals. The entire global network of Networks could be built by interconnecting only Buffer Networks, as long as at least one Buffer Network has at least three Portals. Interestingly, this places an additional burden on the Buffer Network that is not otherwise required; it must have some means of making its own choices of Terminal Nodes, since there is no longer a neighboring Region to make the choice. This author does not advocate this strategy, for several reasons:

a) Many-Portal Regions with relaxed fault recovery capabilities (compared to Buffer Networks) are cost-effective.
b) If adding UNIs also adds Nodes to the Buffer Network, the number of Segments required to provide the necessary guarantees for connectivity are greatly increased.
c) A Buffer Network with only two Portals and no UNIs by definition handles only point-to-point traffic flows, even if the Services carried are multipoint Service. As a result, a Buffer Network never needs to deal with the

customers' addresses; it does not need to learn customer MAC addresses, route customer IP addresses, or filter multicast addresses. Adding a UNI introduces the requirement to perform such functions.

### 5.15 Duplicate and out-of-order delivery

Some packet delivery methods, e.g.Virtual Bridged Local Area Networks (Q-tagged bridging, [2]) provide an extremely low probability of duplicate or out-of-order delivery of customers' packets. Other methods, e.g. IP routing, endeavor to deliver each packet exactly once and in the order transmitted, but make no guarantees, and delivery anomalies must be expected by protocols running over such methods.

Whenever a Region changes the NNI selection on a Portal, there are two opportunities for out-of-order delivery of a stream of packets, or the duplicate delivery of a multicast (or flooded unicast) packet. One occurs when the source NNI changes, i.e. when the Region making the change first transmits on the newly-selected NNI, and the other when the other Region(s)' Terminal Nodes switch paths to respond to the change. If we rule out placing a (perhaps additional) sequence number in every packet carried across the NNI as an unacceptable imposition on the data plane, we are forced to define a "flush" mechanism.

Flushing per-Service is not practical; flushing is done per Bundle. If some, but not all, Services crossing a Buffer Network require protection against duplicate and out-of-order delivery, then an increase in the number of Bundles is required, so that flushing can be accomplished on a per-Bundle basis.

Two flushing mechanisms are required for the two out-of-order opportunities:

a) A Terminal Node A in a Region can transmit a "flush Bundle" message across the NNI. This message can be piggybacked on an NNI change announcement. The Buffer Network Terminal Node B that receives the "flush Bundle" message translates it into one or more "flush Segment" messages, according to the mapping of that Region Bundle in to Buffer Network Bundles, and hence Segments. Terminal Node B then ceases to transmit data for the flushed Bundle. Those flush messages are transmitted to the opposite Terminal Nodes C, D, etc. at the lowest delivery priority of any of the data being flushed. The receiving Terminal Nodes generate flush responses, which are collected by Terminal Node B. When all have been received, it returns a response across the NNI to Terminal Node A.

NOTE—This description assumes that Terminal Node A stops transmitting the Bundle's data after the "flush Bundle" message is sent, and that some other Terminal Node in the Region's side of the Portal resumes transmission after the flush response is received, or after a suitable timeout expires. How this is accomplished is beyond the scope of this document.

b) WhenTerminal Node C in a Buffer Network receives an NNI change notification from Terminal Node B in the Buffer Network, attached to another Region, it must change its selection of Segments for the moved Buffer Network Bundle(s). For each Segment on which a protected Bundle is carried, Terminal Node C transmits a "flush Segment" message, and ceases transmission of the affected Bundle on the old Segment. When the response is received (or a timeout expires), Terminal Node C resumes transmission of the affected Bundles on the new Segment.

The same "flush Segment" mechanism can be used when only a single Service is affected, as for example when the Bundling is reconfigured, by impeding only the transmission of that Service.

If the protection-switched Buffer Network of section 5.4 is implemented, then the "flush Segment" message and response can be the Loopback Message (LBM) and Loopback Response (LBR) of [3] and [4]; no new control packet needs to be defined. If Terminal Pairs, rather than physically separate Terminal Nodes and connecting NNIs, are employed, then the "flush Bundle" message is internal to the Terminal Pair chassis, and need not be defined as an explicit control packets.

### 5.16 Dynamics

Networks are not static; new Nodes and Links are added to Networks, and old ones are replaced or taken down temporarily for maintenance. If Buffer Networks are implemented, and use a prioritized repertoire of pre-configured

Segments as outlined in section 5.4 and the protocol enhancements suggested by section 5.16 are made, then it is very straightforward to sequence changes to the configuration of a Buffer Network to allow, for example, the addition of a new Terminal Pair, followed by the removal of an existing Terminal Pair, with the final result that the Terminal Pair can be replaced without ever leaving the global network with sub-standard fault recovery capabilities.

Adding or deleting Services and/or UNIs can result in changes to the Bundling of Services. When such a change take place, the inevitable propagation delays result in different Nodes affecting the change at different times. In the protection-switched Buffer Network described in section 5.4, this presents no problem with global forwarding loops or basic data delivery. A temporary lapse of Service congruency (section 3, item c) is therefore likely. Packets may also be duplicated or delivered out of order, unless care is taken as described in section 5.15.

### 5.17 Protocol elements

At this stage, there are many open questions. The following primitives may be useful to make Buffer Network Insertion work. (Note that primitives that flow over and NNI that is embedded in a Terminal Pair is not visible outside the chassis of the Terminal Pair.)

a) A means of trading identity information before, and periodically during (for safety). The identity traded must be sufficient to ensure that both of the Networks agree on which NNIs belong to which Portals, that there are only two Networks involved with the Portal.
b) If a Buffer Network is virtualized, in the sense that all of its Terminal Nodes belong to Terminal Pairs, then the identity information mentioned in item (a) must be exchanged among the Terminal Pairs across the Buffer Network to ensure that the right Regions are interconnected.
c) The two Terminal Nodes across an NNI must be able to signal their ability or inability to exchange a Bundle of Services.
d) A signal from the Region to the Buffer Network over an NNI saying, “The following is the order of preference for the NNI to be used for this Service or Bundle of Services” may be useful.
e) A signal from the Region to the Buffer Network over an NNI saying, “This NNI can be used for the following Bundles and/or Services” is required. (The most-preferable NNI according to the list in item (d) is used.)
f) The Bundling choices made by the Region must be conveyed to the Buffer Network if the signal in item (d) or item (e) can carry a Bundle identifier.
g) The Bundling, if any, used by the Buffer Network, can be conveyed by the control path, rather than by configuration. (See also section 5.11.)
h) The NNI choices made by the Regions must be conveyed to all Terminal Nodes in the Buffer Network. This can be done either by relaying the signals from the Regions [item (c), item (d), item (e)], or by summarization.
i) It is for further investigation whether signals from the Region to the Buffer Network over an NNI saying any of the following will be useful, or simply add complexity:
    1) “This NNI can *not* be used for the following Bundles and/or Services”
    2) “This is the right NNI for the following Bundles and/or Services.”
    3) “This is *not* the right NNI for the following Bundles and/or Services.”
    4) “NNI <id> is now the right NNI for the following Bundles and/or Services and all others are not.”
j) “Flush Bundle” messages across the NNI and “Flush Segment” messages across the Buffer Network are required if and Services require protection against out-or-order or duplicate delivery.

## 6. References

[1] Finn, Norman, “A framework for defining an IEEE 802.1 NNI”, presentation to IEEE 802.1, January 2010, http://www.ieee802.org/1/files/public/docs2010/new-nfinn-nni-framework-0110-v01.pdf.
[2] IEEE Std™ 802.1Q-2005, “Virtual Bridged Local Area Networks”, http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf.
[3] IEEE Std™ 802.1ag-2007, “Connectivity Fault Management”, http://standards.ieee.org/getieee802/download/802.1ag-2007.pdf.

[4] ITU-T Recommendation Y.1731(2008), “OAM functions and mechanisms for Ethernet based networks”, http://www.itu.int/rec/T-REC-Y.1731-200802-I/en.

[5] Nadeau, T. and R. Aggarwal, “Pseudo Wire Virtual Circuit Connectivity Verification (VCCV)”, RFC 5085, December 2007, http://www.ietf.org/rfc/rfc5085.txt.

[6] IEEE Std™ 802.3-2005, “Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications”, Clause 43 Link Aggregation, http://standards.ieee.org/getieee802/download/802.3-2005_section1.pdf.

[7] Atlas, A., Ed., Zinin, A., Ed., “Basic Specification for IP Fast Reroute: Loop-Free Alternates”, RFC 5286, September, 2008, http://www.ietf.org/rfc/rfc5286.txt.

[8] IEEE Std™ 802.1ah-2008, “Virtual Bridged Local Area Networks Amendment: Provider Backbone Bridges”, http://standards.ieee.org/getieee802/download/802.1ah-2008.pdf.

[9] Haddock, Stephen, “E-NNI Redundancy”, presentation to IEEE 802.1, November 2009, http://www.ieee802.org/1/files/public/docs2009/new-haddock-ENNI-redundancy-1109-v1.pdf.

[10] IEEE Std™ 802.1ak-2007, “Virtual Bridged Local Area Networks Amendment: Multiple Registration Protocol”, http://standards.ieee.org/getieee802/download/802.1ak-2007.pdf.