Comparing LACP and Buffer Networks

Norman Finn

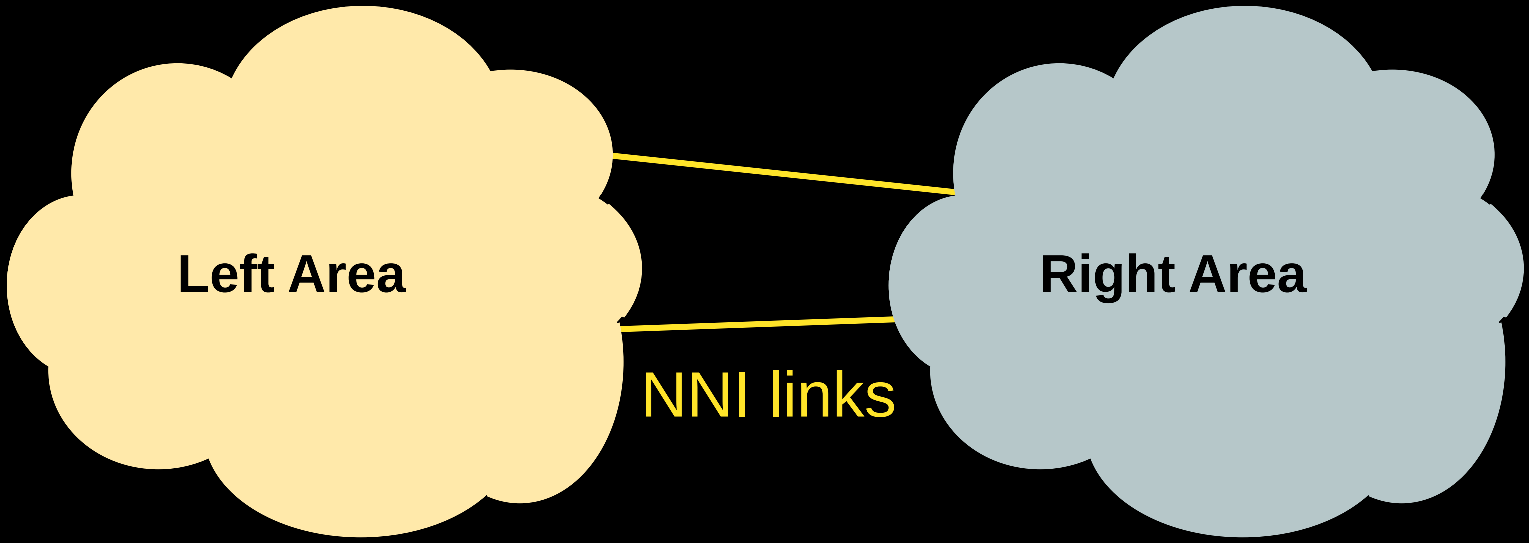
Rev. 1

Comparing LACP and Buffer Networks

- This document is new-nfinn-LACP-vs-buffer-networks-1110-v1.pdf.
- Summary:
  1. There is no “heavy” or “light” solution; the number of components and links and the flow of data frames are driven by the problem requirements, and will be the same, whether we select an LACP-based solution or some other solution.
  2. We have a choice between bridging/routing technology or protection switching technology for the data plane.
  3. After that, the requirements for what control information must be either statically configured or passed through the control plane can be met by several protocols.

Problem statement

Problem Statement



- We want to connect two independent Ethernet Service Providers’ clouds (let’s call them “Areas”) with some number of Network-Network Interfaces (NNIs) to provide redundancy and load sharing.

802.1 has an ambitious set of requirements for NNIs

1. Failure or recovery action in one Area **never** triggers an action in an adjacent Area.
2. Areas may bundle services for scaling purposes (e.g. failure recovery) totally independently.
3. Load sharing of NNIs is necessary.
4. Fast failover is required.
5. Areas may use different failure recovery methods, say, 802.1aq SBP vs. .1Qay PBB-TE.
6. More than two nodes or two links must be supported, so that full protection can be maintained while replacing equipment.
7. Solution must not require ultra-dependable links.
8. Solution must provide a means to not increase the chance of duplicate or out-of-order packet delivery.
9. We must support least 802.1ad and 802.1ah networks.

Non-requirements

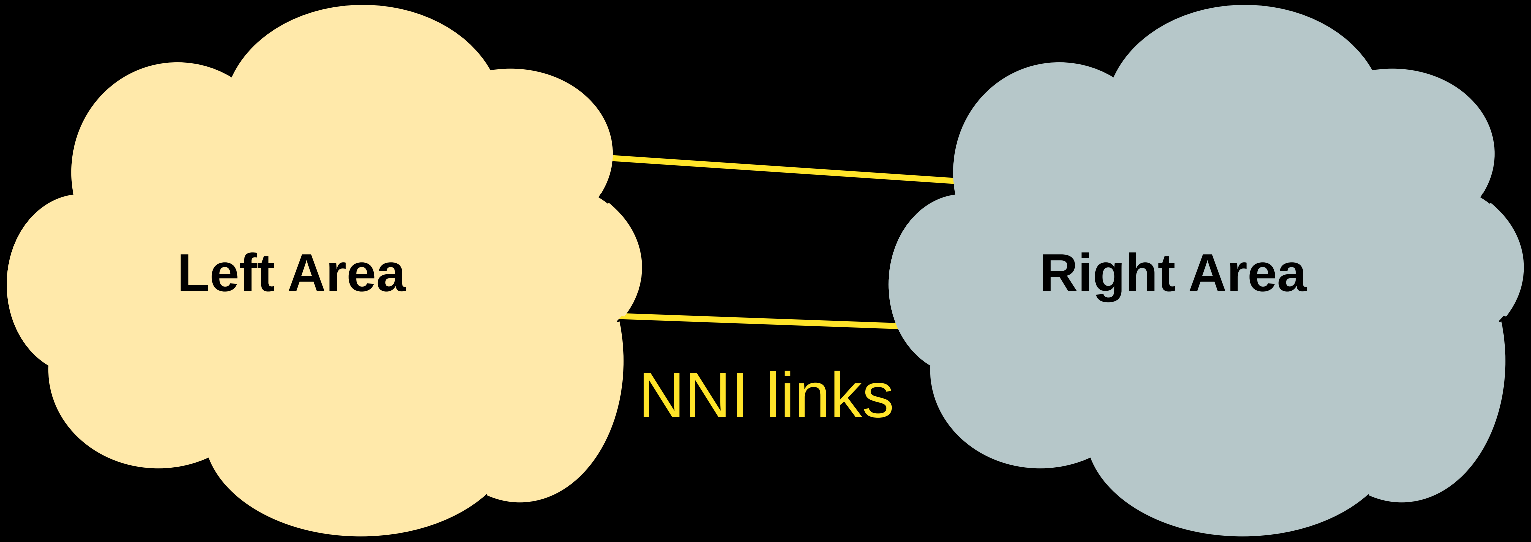
- Two **non-requirements** are also important:

1. If an Area is split, adjacent Areas will **not** provide connectivity.
2. Only connections between **pairs** of Areas need be considered.

- Together, these non-requirements mean that the interconnect never deals with MAC addresses or multicast distribution trees, which greatly simplifies its interactions with the Areas.
- Of course, MAC address awareness could be added at some point.

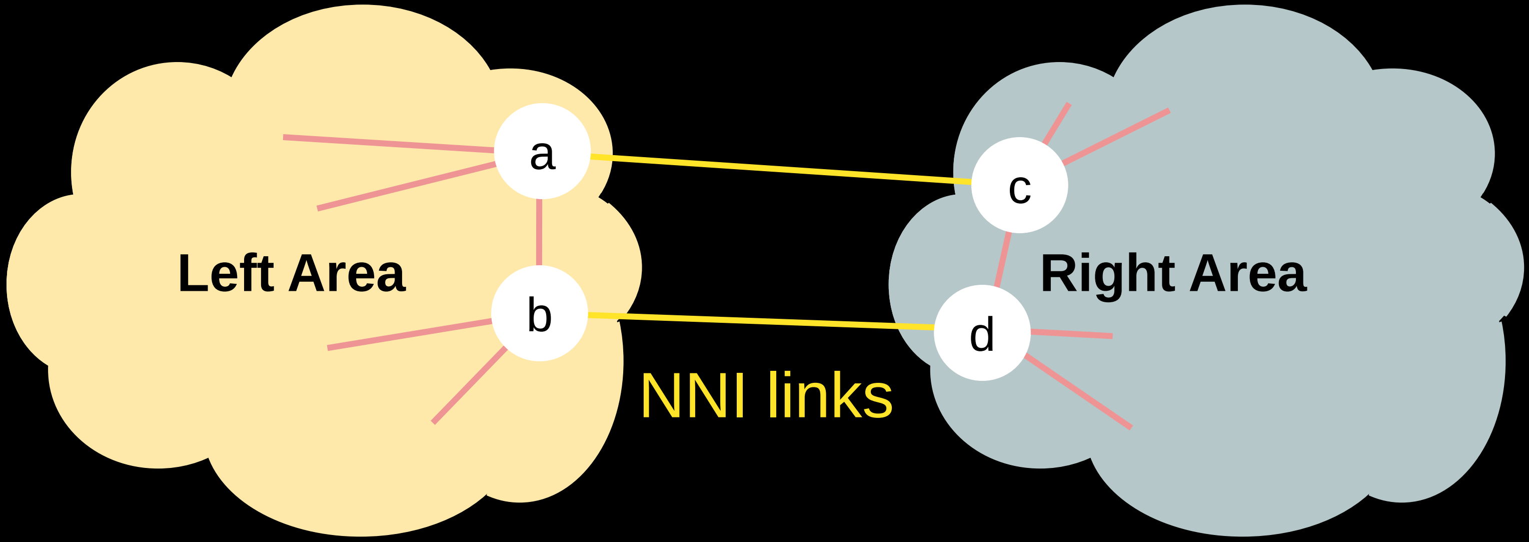
Buffer Network model

Building a Buffer Network



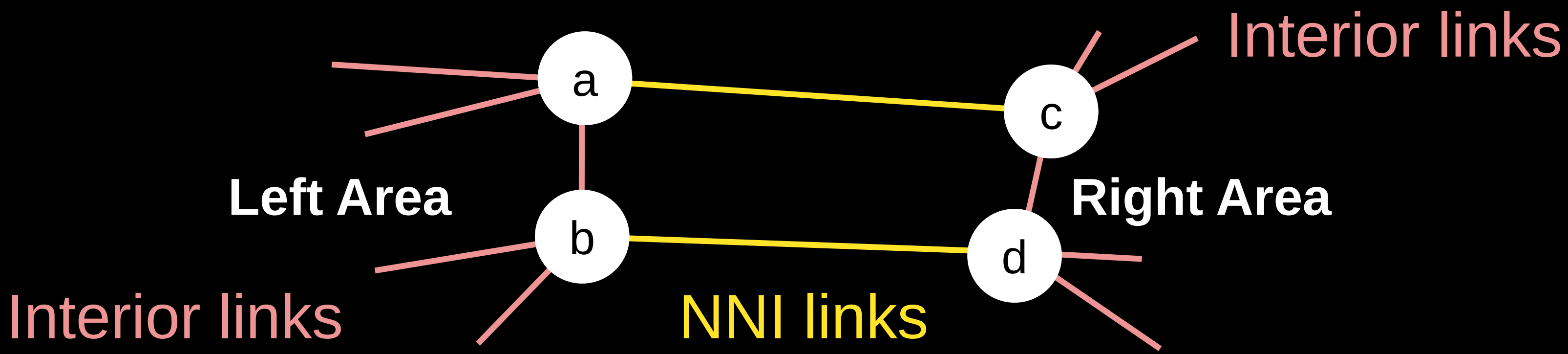
- Let us zoom in on the devices

Building a Buffer Network



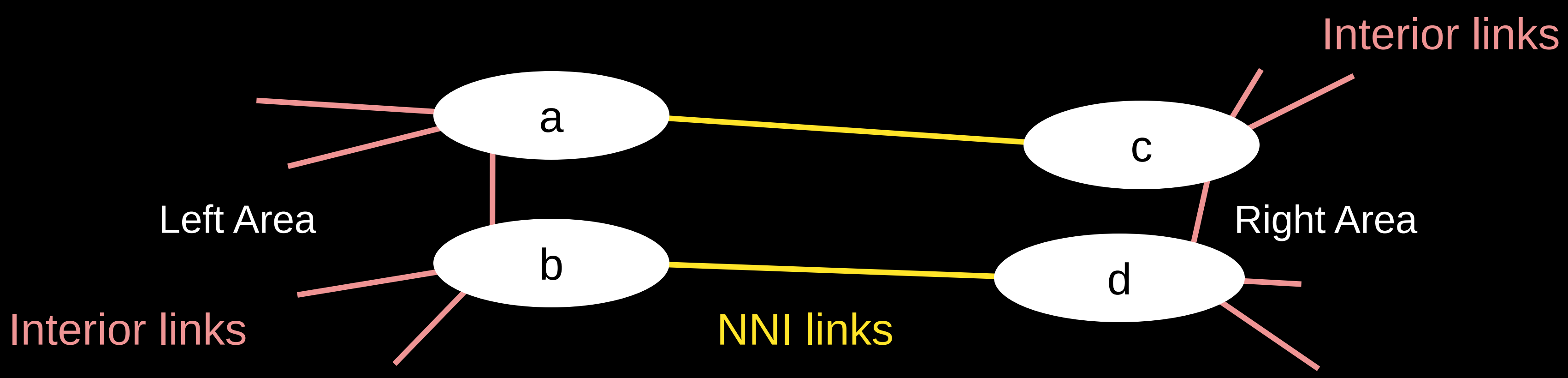
- Let us zoom in on the devices

Building a Buffer Network



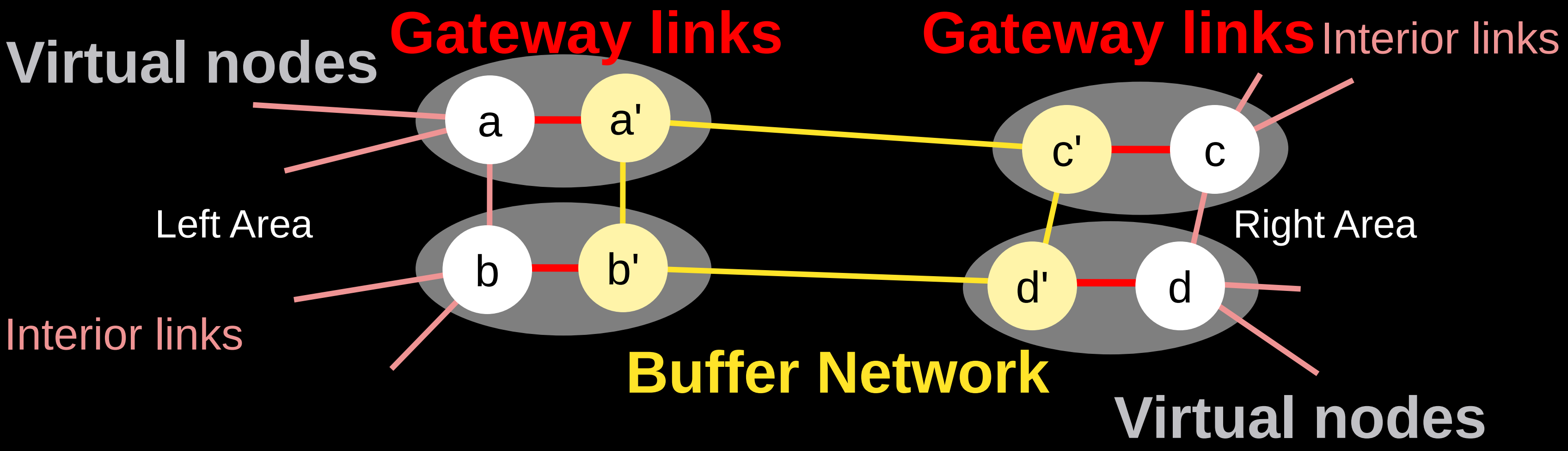
- Let us zoom in on the devices

Building a Buffer Network



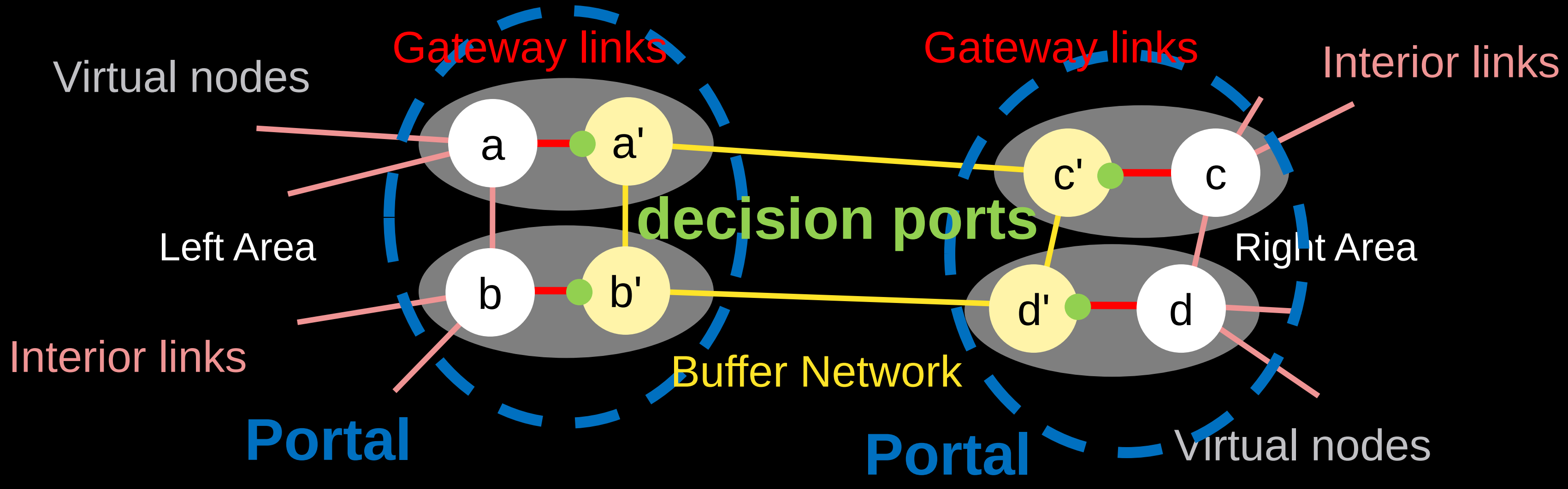
- Let us morph the devices

Building a Buffer Network



- We have split each bridge into two components.
- We require links between a'-b' and between c'-d', whether they are physical or logically shared with the a-b or c-d links.
- We now have a **Buffer Network**.

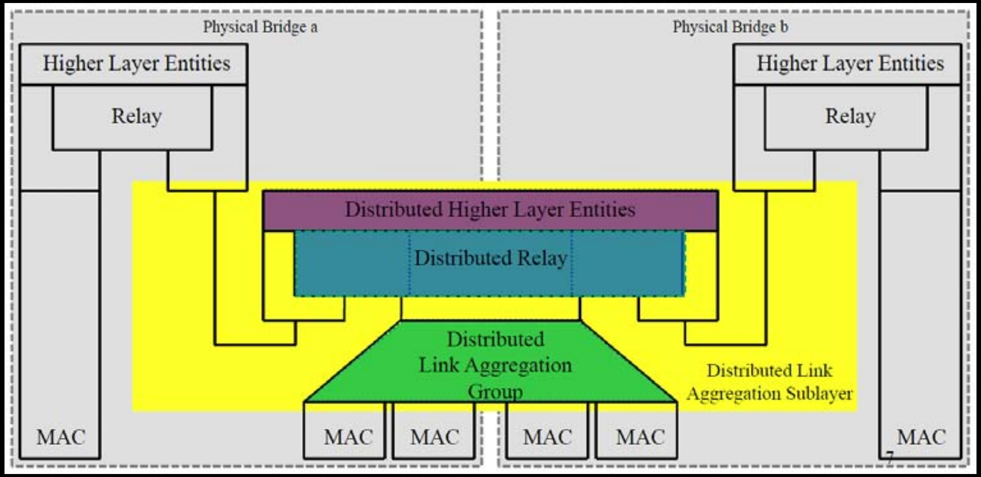
Building a Buffer Network

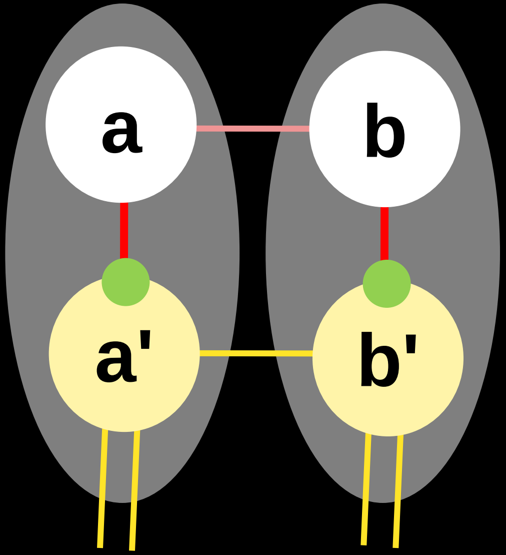


- The Buffer Network is jointly operated by the two Providers; we will make it as simple as possible.
- The Gateway links and decision ports are (usually) internal to a physical box, so are invisible to the outside world.

Distributed LACP vs. Buffer Networks

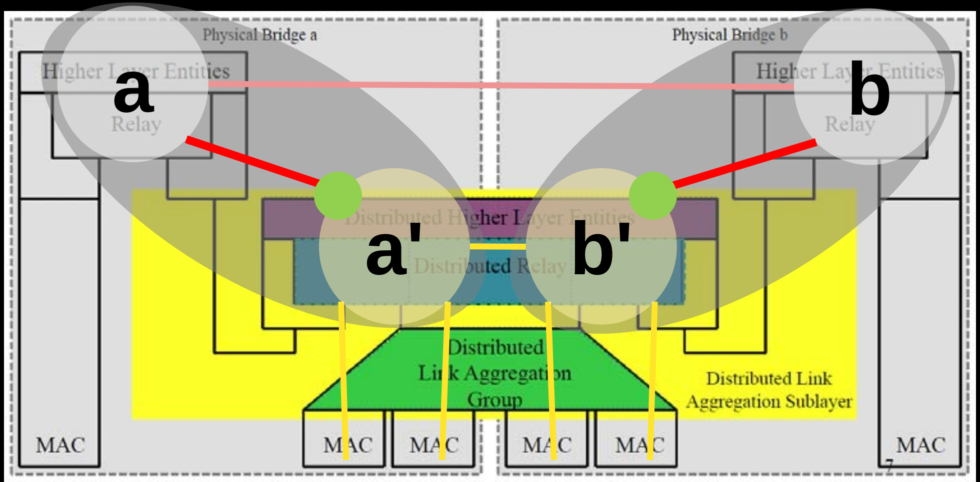
Distributed LACP vs. Buffer Networks





- The left-hand model is taken from new-haddock-Distributed-LAG-Models-1010-v2.pdf.
- **Q:** What is the difference between these two models?

Distributed LACP vs. Buffer Networks



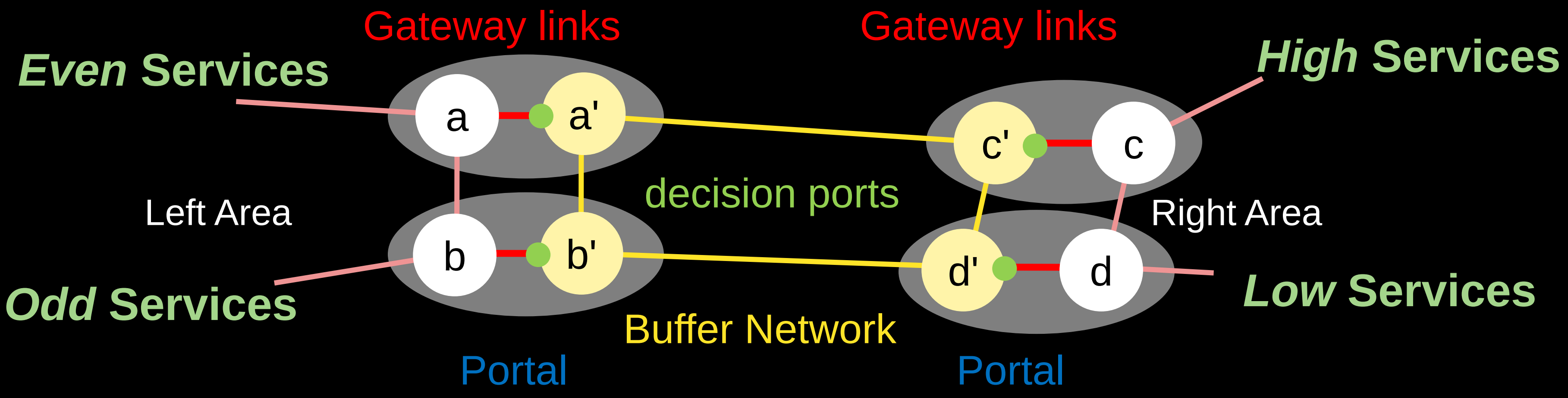
- **A: None**, in the data plane.
- Every data frame must travel along the same paths, with the same restrictions, whether we emulate a distributed bridge or a buffer network.
- Only the names of the components, and potentially the choice of tags, can vary.

For **both** models:

- Exactly one Gateway link (physical bridge to distributed relay link) among all those connected to the Buffer Network (distributed relay) carries all of the services belonging to a given B-VID (S-VID), else the Area can suffer from address flapping.
- There must be a data path among the nodes within a portal (the physical parts of the distributed relay) in order to reconcile the different bundling plans used by the two Areas.
- There **are** choices to be made with regard to tagging, and these choices influence **what protocol** runs among the Buffered Network nodes (distributed relays).

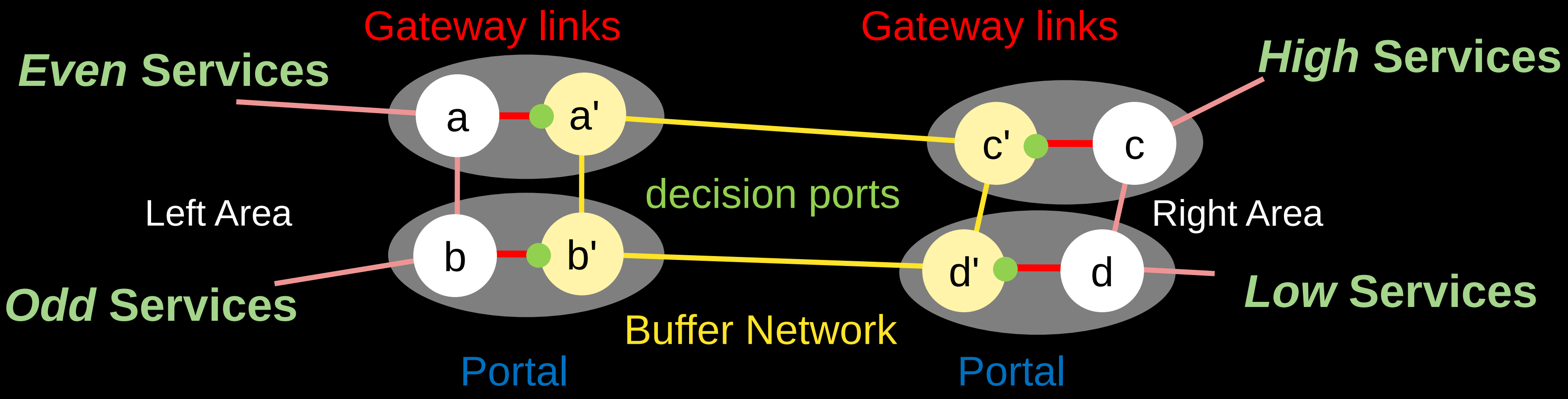
Bundling requirements

Bundling



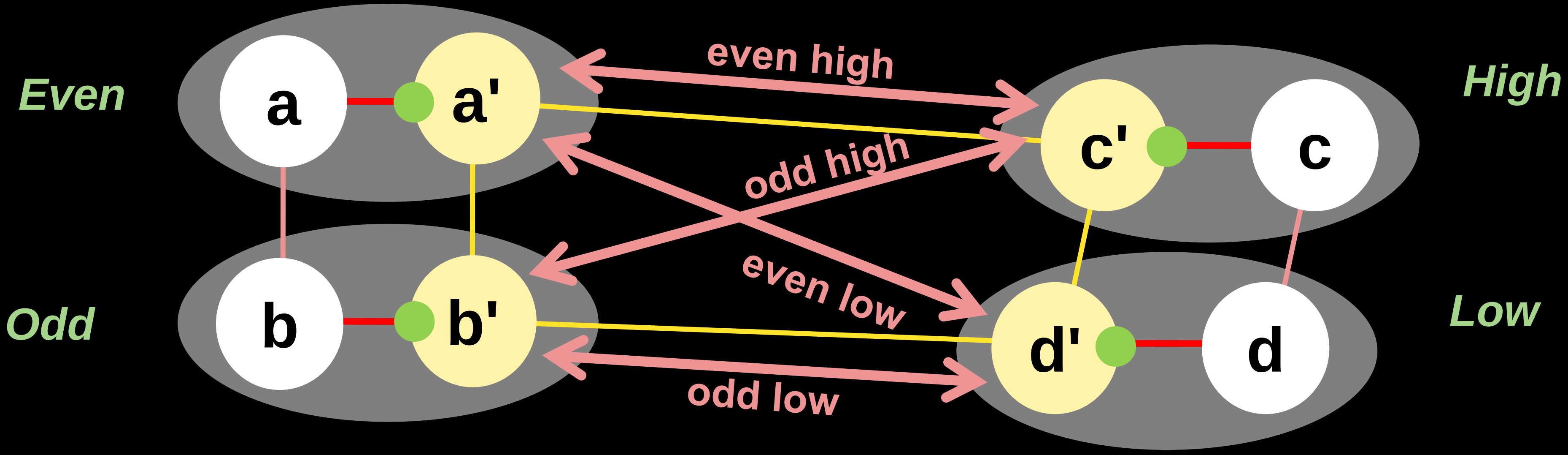
- (I use Buffer Network terminology, but DLACP terminology is equally applicable.)
- For scalability, each Area very likely groups Services into **Bundles**. (There are too many I-SIDs to signal them individually.)
- But, bundling is **different** in the two Areas.

Bundling



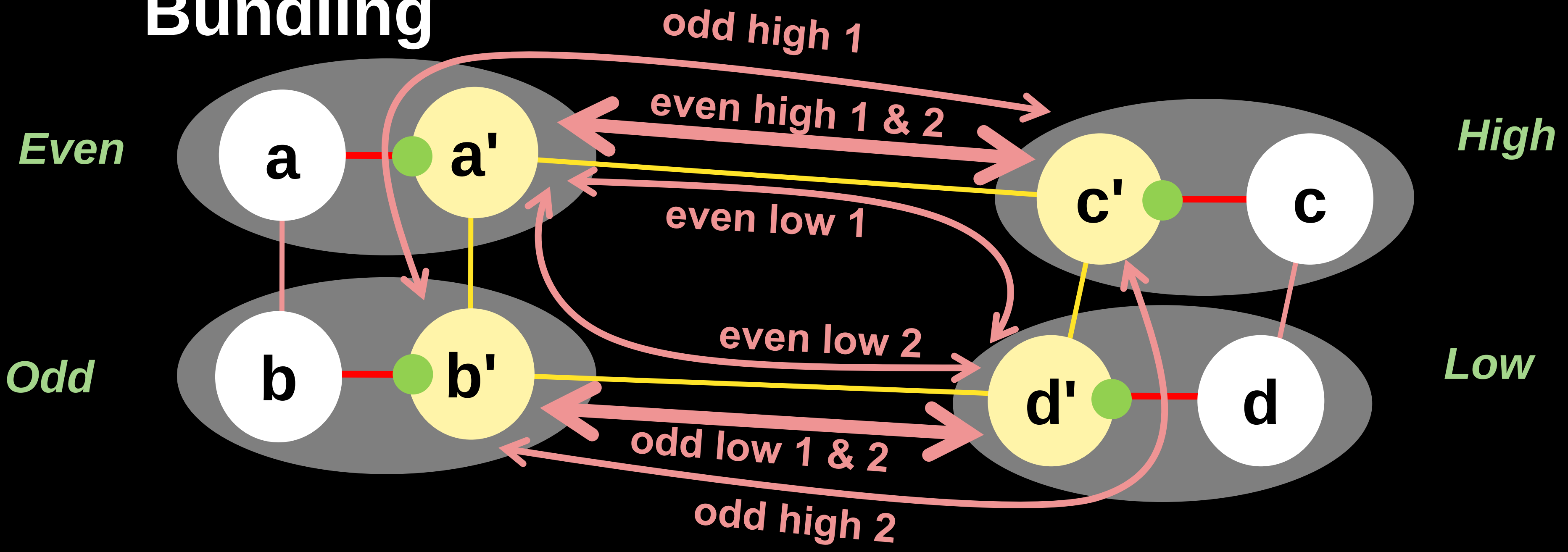
- For each Portal, either the Area or the Buffer Network can be selected by configuration to be in charge of assigning each Bundle of Services to exactly one Gateway link.
- (Why? Some Area protection protocols like to make the choice, and some do not.)

Bundling



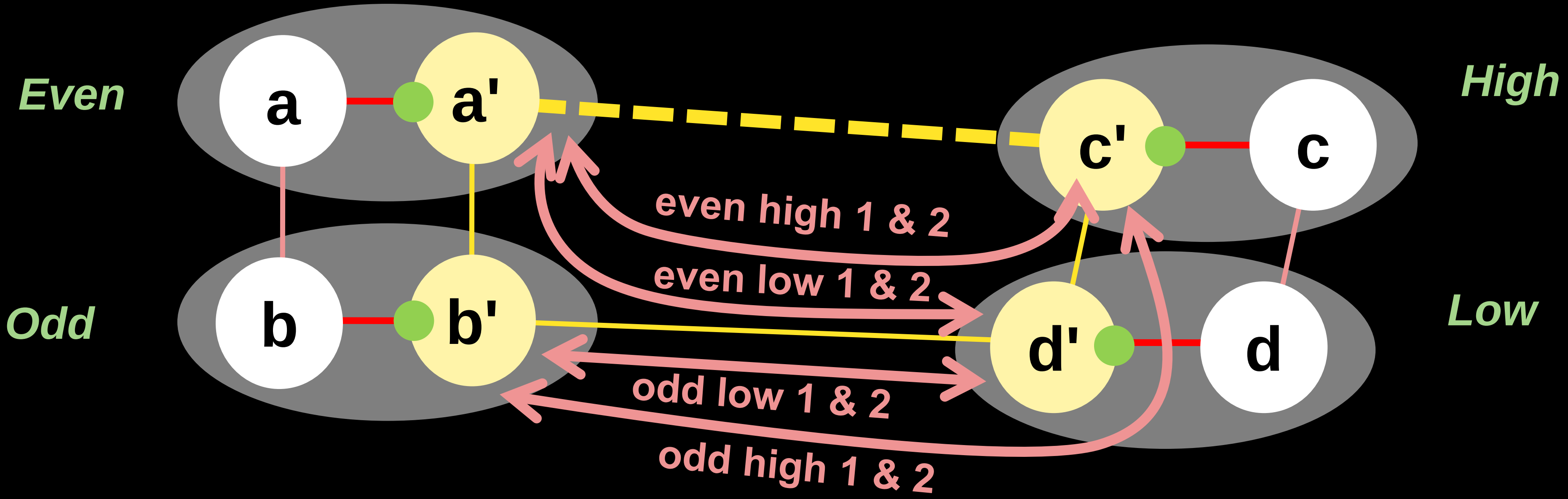
- The Services are bundled, either by jointly-agreed configuration or protocol action, in the Buffer Network.
- In general, more Bundles are needed in the Buffer Network than in either of the Areas.

Bundling



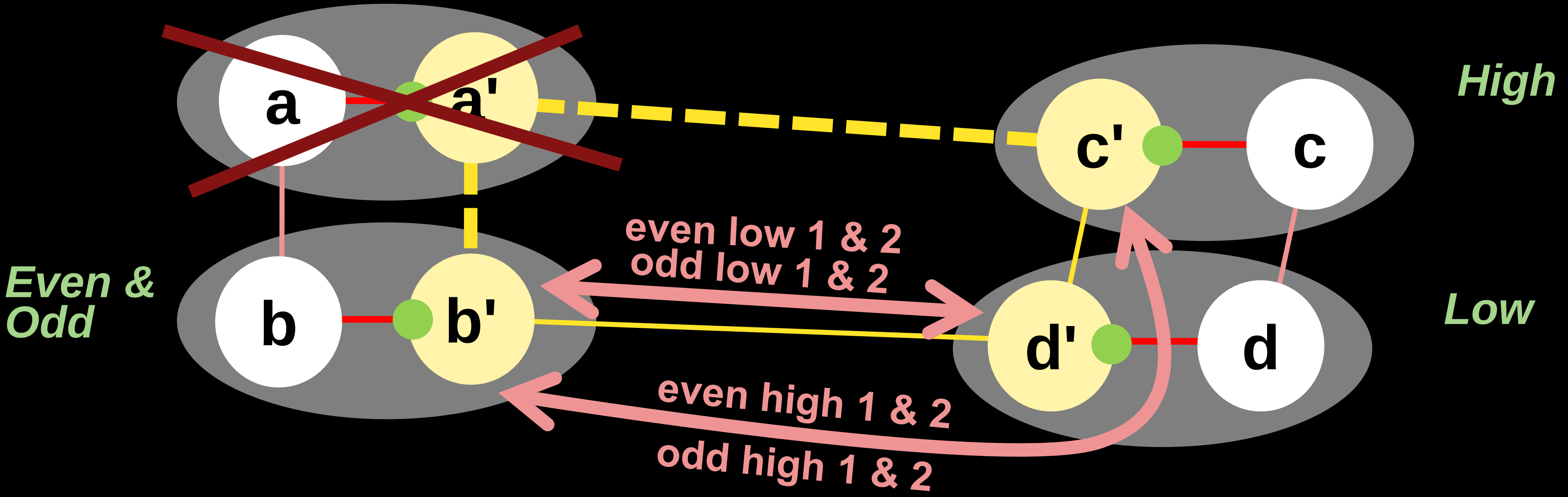
- **Eight** bundles are required in this example, because both of the a'-d' and both of the b'-c' paths must be used in order to load-balance the a'-b' and c'-d' links.
- Note that, if the eight Bundles are equal in required bandwidth, the links are perfectly shared.

Bundling and failure protection



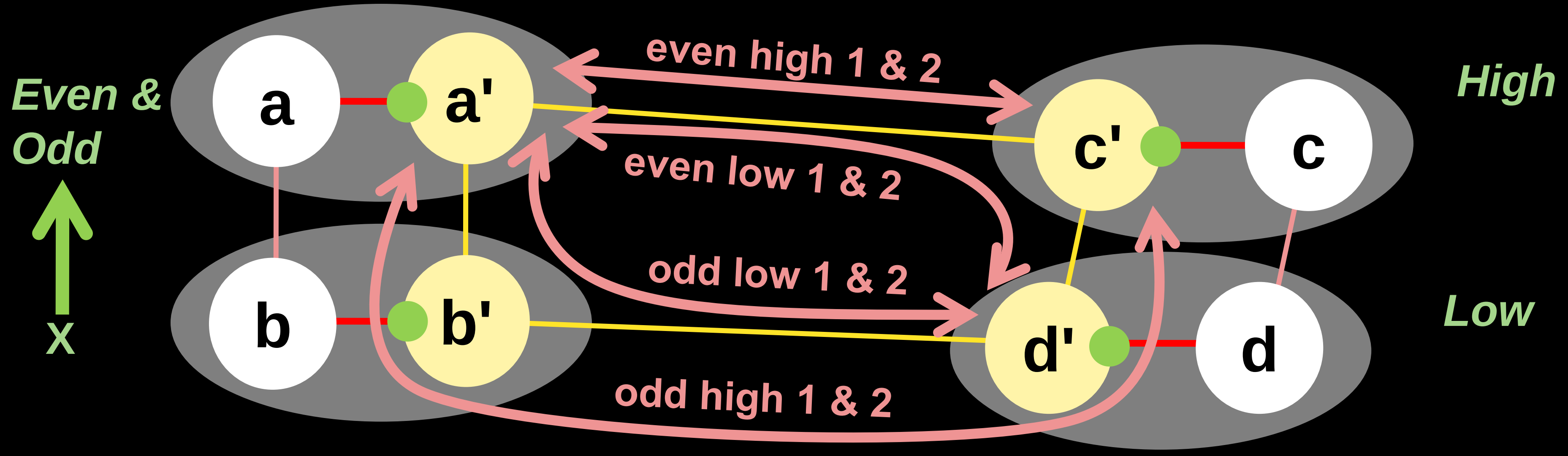
- If anything happens to a link (e.g. a'-c') in the Buffer Network, the Buffer Network redistributes the load, and the **Areas are not affected**.
- (Both providers' boxes are affected, but only the parts belonging to the Buffer Network – not the parts participating in the Areas' control protocols.)

Bundling and failure protection



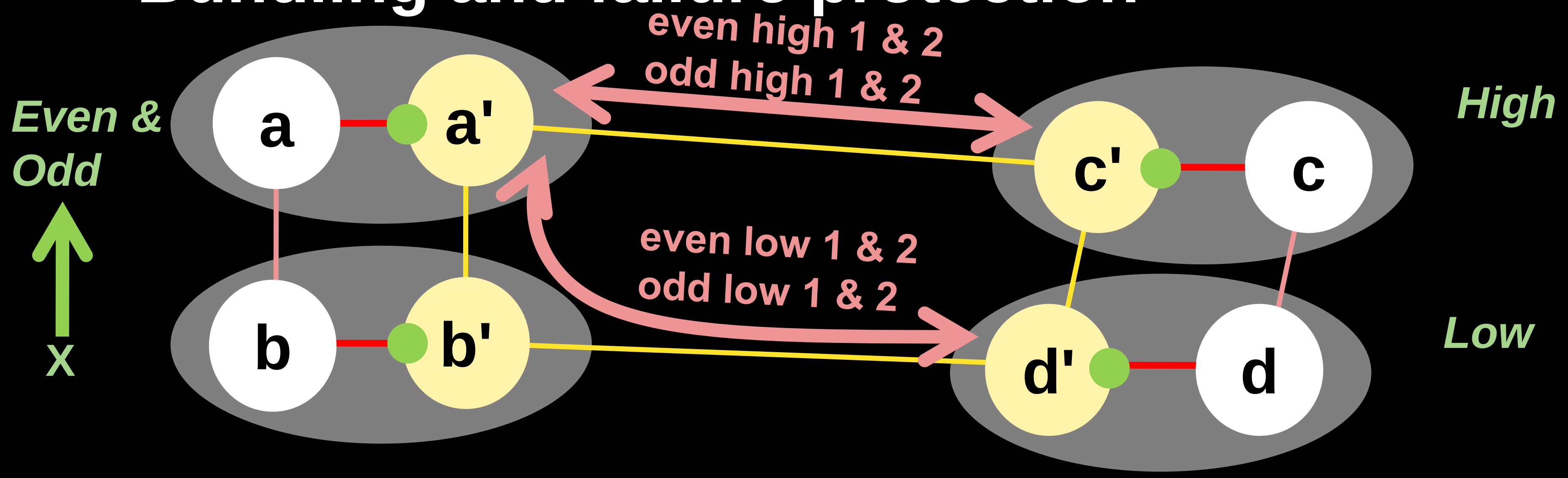
- If anything happens to a Node (e.g. a-a') in the Left Area, the Buffer Network and the Left Area reroute the Bundles and the **other Area is not affected**.

Bundling and failure protection



- If the Left Area changes its mind about load sharing, the Buffer Network adjusts, and the **other Area is not affected**.
- This arrangement optimizes load sharing at the expense of latency (the Odd-High Bundle takes the long route).

Bundling and failure protection



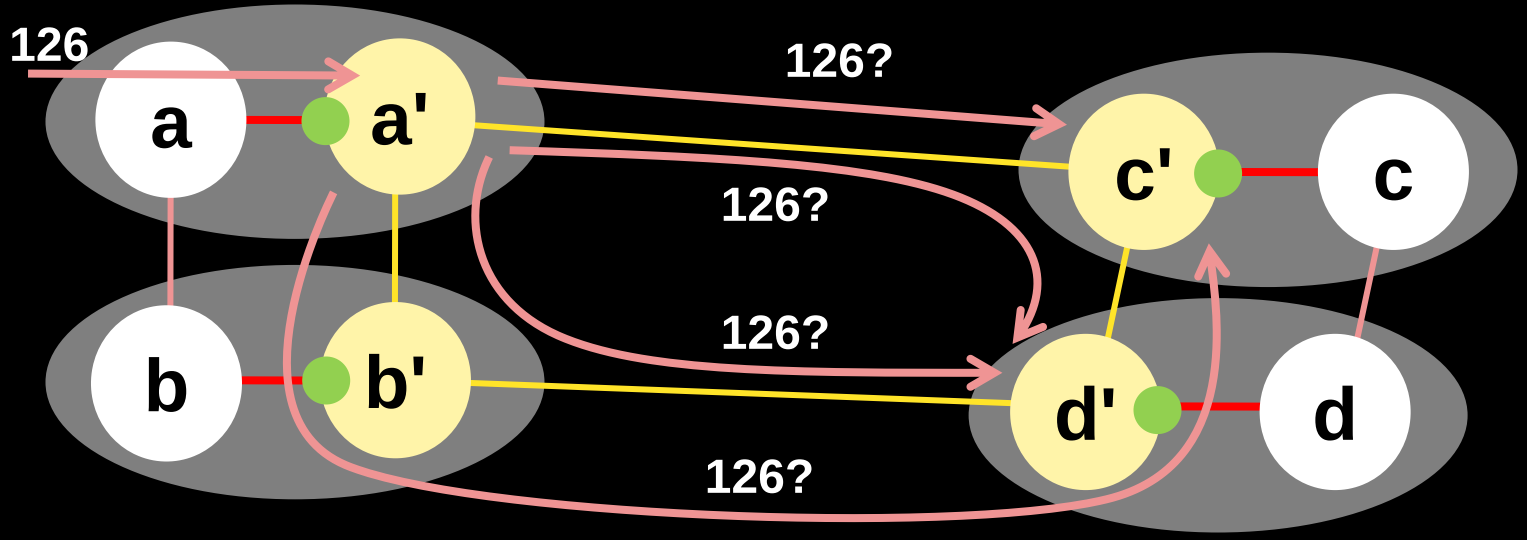
- Same condition (left Area changes Bundle-to-Gateway assignments), different answer.
- This arrangement optimizes latency at the expense of uneven load sharing (a'-b' carries half the load, and c'-d' none).

Routing/bridging versus protection switching

Routing vs. Protection Switching

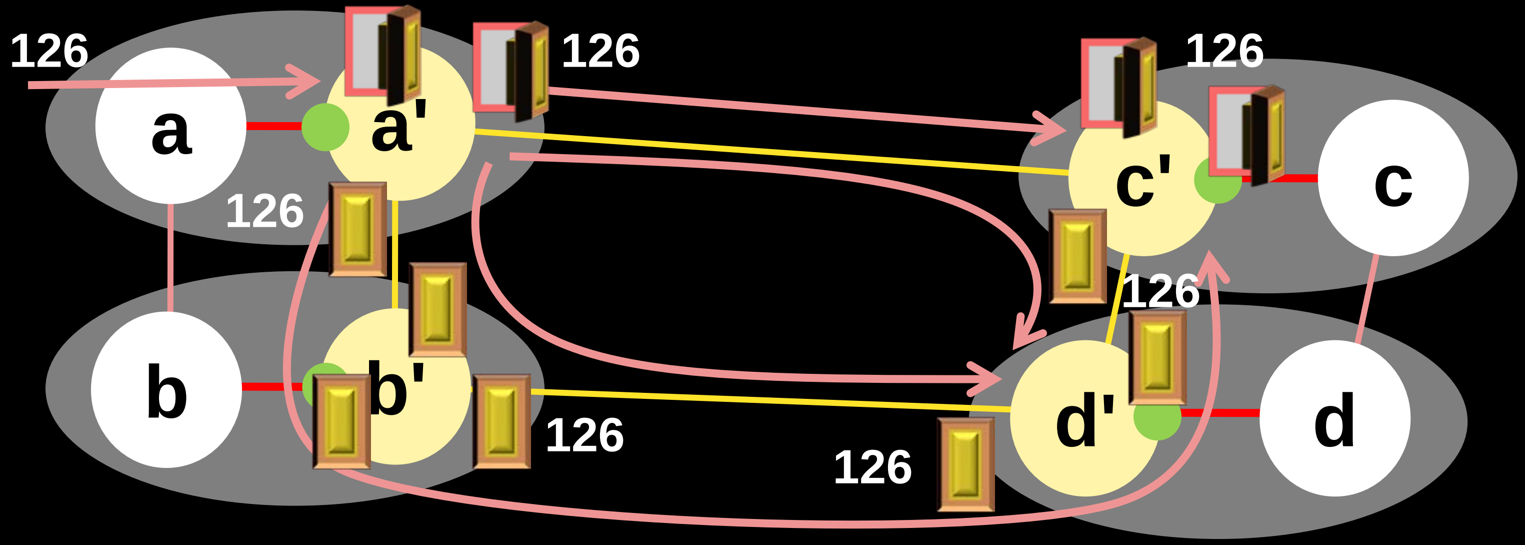
- We must make a decision whether to use protection switching technology in the interconnect or to use routing/bridging technology.
- In both methods, complete knowledge of the state of the interconnect network is required for all nodes to make the right decision to effect connectivity and balanced load sharing.
- By “protection switching technology” we mean that each frame is assigned to a pre-configured tunnel as it enters the buffer network, and either discarded or delivered when it emerges on the other side.
- By “routing/bridging technology” we mean that each node makes an independent decision as to which port to transmit each frame, based on the frame’s service ID.

Routing vs. Protection Switching



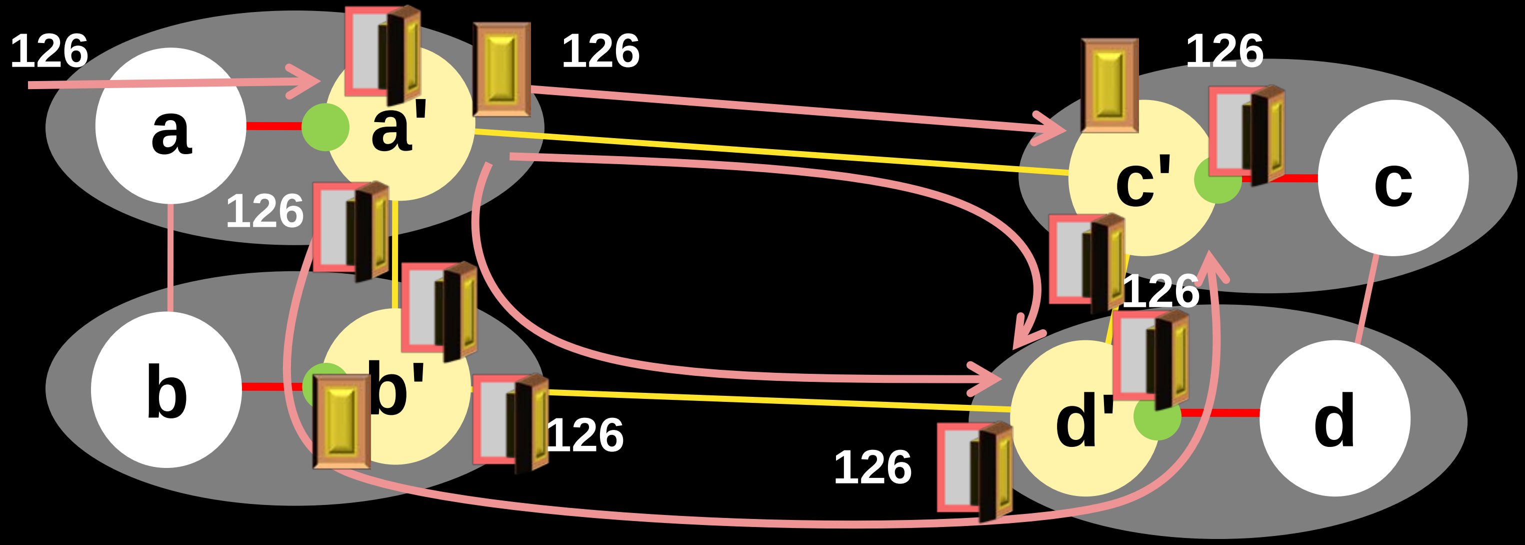
- Depending on the state of the Areas and the Buffer Network, a frame tagged with “Service 126” entering the a-a' Node could take any of four paths to get to the right-hand Portal.
- **How** is the frame’s path determined in the **data plane**?

Routing



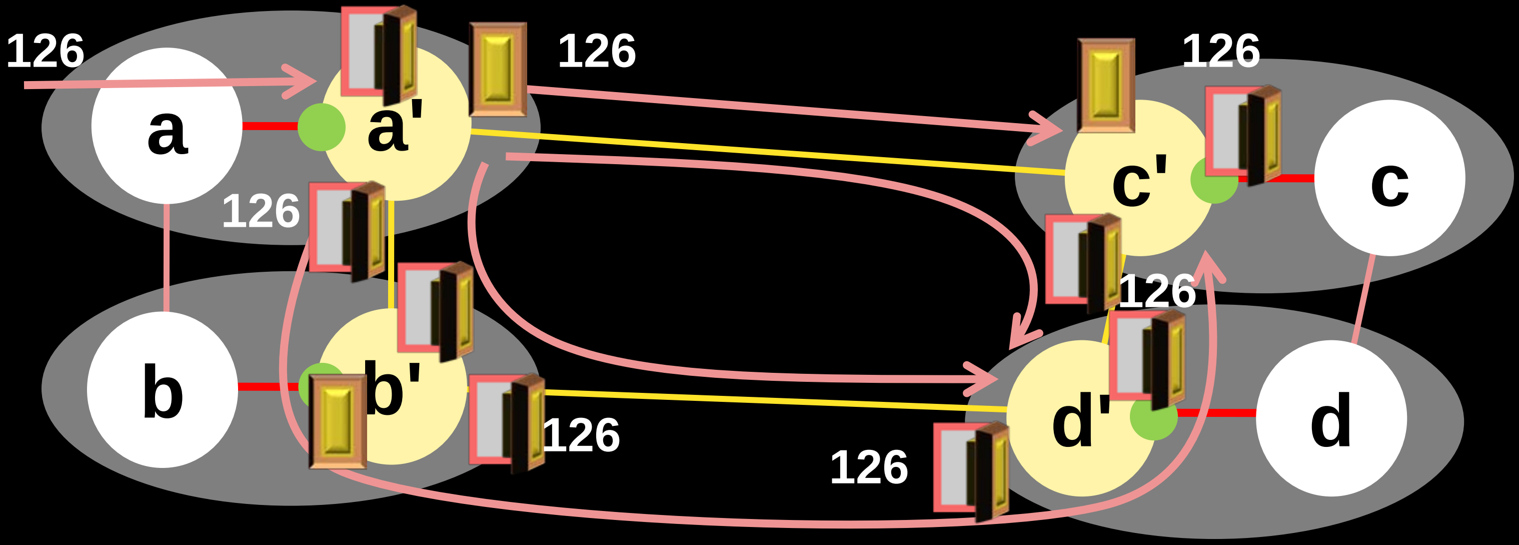
- We can keep the original tag (with perhaps one translation required by differing tag values in the two Areas) and open/close doors to indicate where Service 126 can go.
- (We could also add a tag identifying a Bundle of services. The essential elements here are the doors.)

Routing



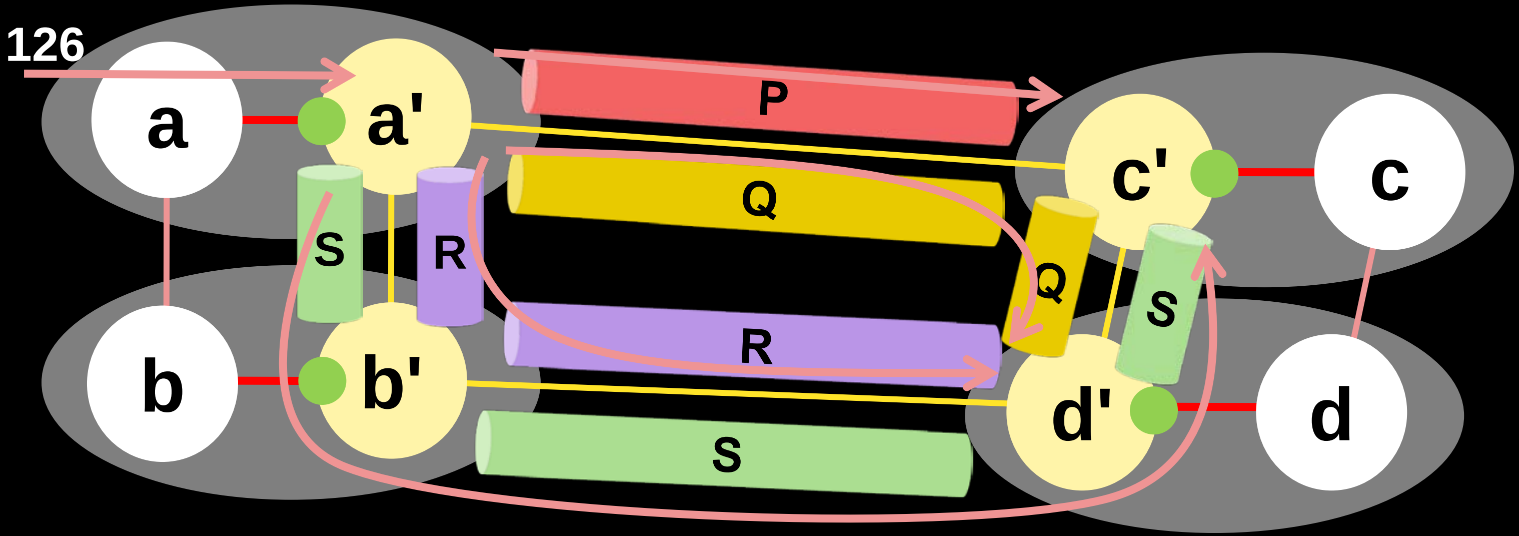
- When we change routes, we have to **open** some doors and **close** others.
- **Multiple Nodes** (perhaps all) must make a **change**.

Routing



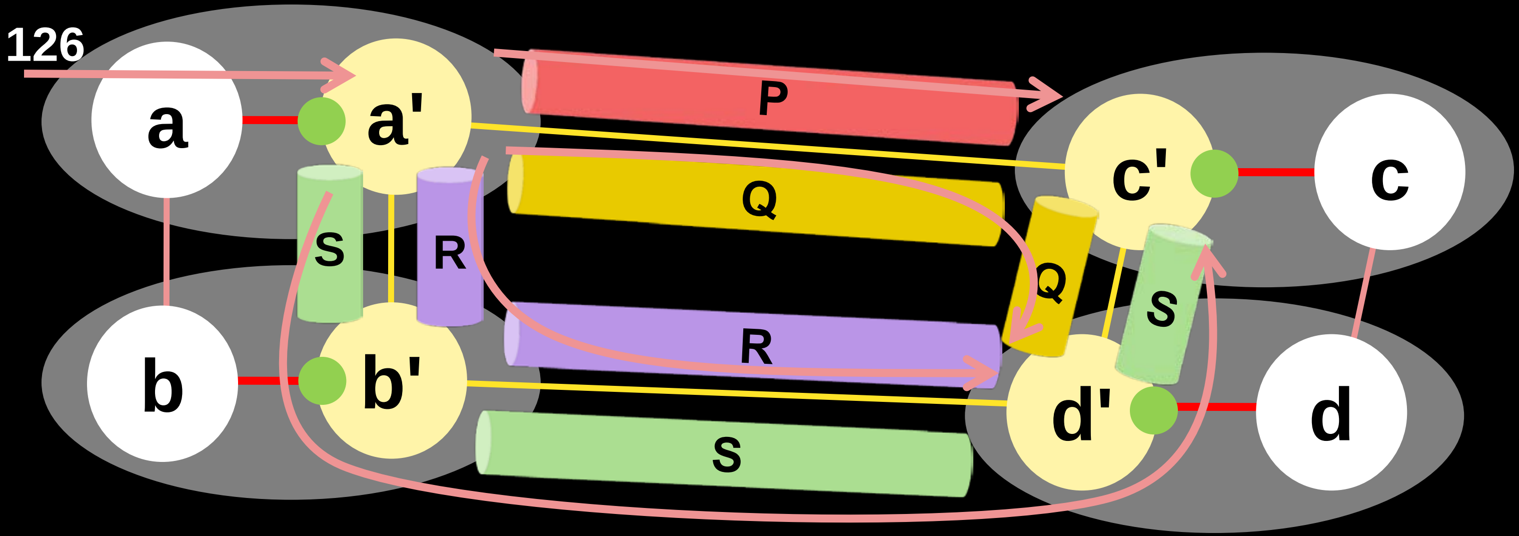
- This requires **interlocks** to prevent forwarding loops. (E.g., switching from a-a'-b'-d'-c'-c to b-b'-a'-c'-c causes loop a-a'-b'-b-a if a' and b' don't shake hands.)
- Note that interlock is only required when **two changes**, b'-d' to a'-c' and a-a' to b-b', take place more or less simultaneously.

Protection Switching



- We build pre-configured tunnels, and Node a' picks one – **P**, **Q**, **R**, or **S**.
- If the situation changes, Node a' picks another tunnel. (The choice changes, not the tunnels!)
- For events occurring within the Buffer Network, **only one Node changes** – **no interlocks** needed.

Protection Switching



- But, we now must either **change** the frame’s encapsulation or **add another layer** of encapsulation, in order to identify which tunnel the frame is taking.

Routing vs. Protection Switching

- There is no news, here! Protection switching can be faster than a bridging/routing protocol, but it requires an encapsulation plan. This is an engineering tradeoff.
- If we use protection switching, it is hard to see why we would use LACP. More likely, we would use some form of 802.1ag CFM or ITU-T Ethernet Protection Switching.
- If we use a bridging/routing protocol, then we could either adapt an existing protocol (LACP? MSTP? SPB? CFM?) or invent a new one.

Protection Switching Encapsulation

- If 802.1ah MAC-in-MAC is used, then the Services are marked by I-SIDs over the Gateway links, the Buffer Network Tunnels are B-VIDs, and the Decision Ports are CBPs. We know how to do that!
- If 802.1ad Q-in-Q is used, then the Services are marked by S-VIDs, and we have a choice of how to mark a Tunnel:
  1. We can use one S-VID per tunnel per Service, and Decision Ports map the service tag to the right S-VID.
  2. A Decision Port can add a “protection tunnel ID” tag, using the original S-VID just like a CBP uses an I-SID.
  3. We can forget protection switching, and change the routes used by the S-VIDs using an interlocked control protocol.

Protection Switching for S-tags

- Protection switching S-tagged services has issues.
- One S-VID per tunnel per Service drastically reduces the number of Services that the Buffer Network can carry (by a factor of at least 7 in the above example), but the Decision Port is an ordinary Provider Network Port.
- Adding an extra tag requires a new kind of Bridge Port (an S-tagged version of a CBP), and opens a Pandora’s box of possibilities.

Guaranteeing in-order frame delivery

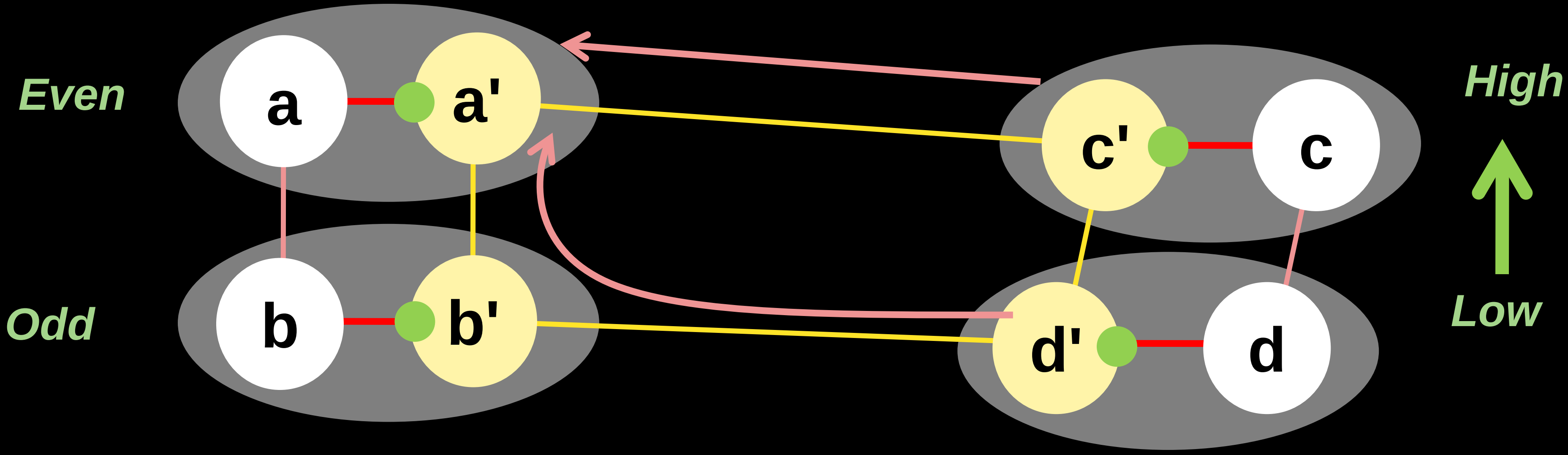
Frame ordering

- Although **guaranteed delivery order is not required by many Providers** (How many vendors and how many network administrators use Link Aggregation’s Marker PDUs?) it would be a shame if a Buffer Network were unable to support two Areas’ abilities to guarantee (or almost guarantee) against duplicate or out-of-order frame delivery.
- Whether we use protection switching or bridging/routing, all chances for frame ordering stem from either:
    1. Changing from one path to another within the BN.
    2. Changing from one Gateway to another in a Portal.

Frame ordering: Path to path

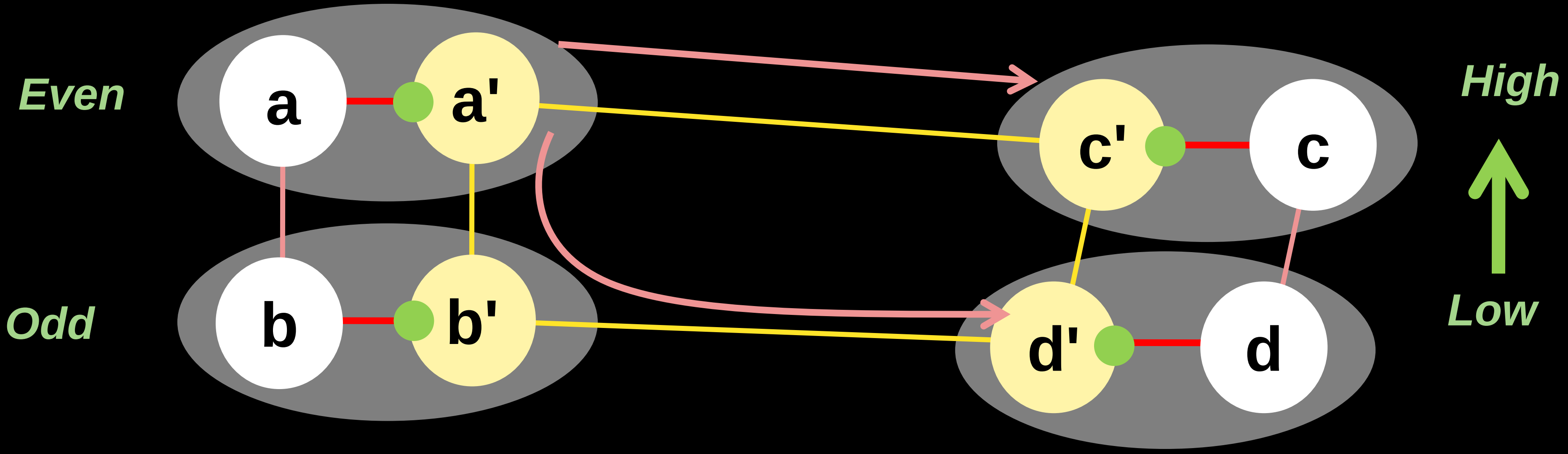
- As long as both ends of a path block unexpected Bundles on that path and enable only one path at a time for transmission (i.e., “break before make”), there can be no frame ordering issues for path changes within the Buffer Network.
- This technique requires **no interlocking** (hand shaking) among the Decision Ports of a Portal.

Frame ordering: Gateway to Gateway



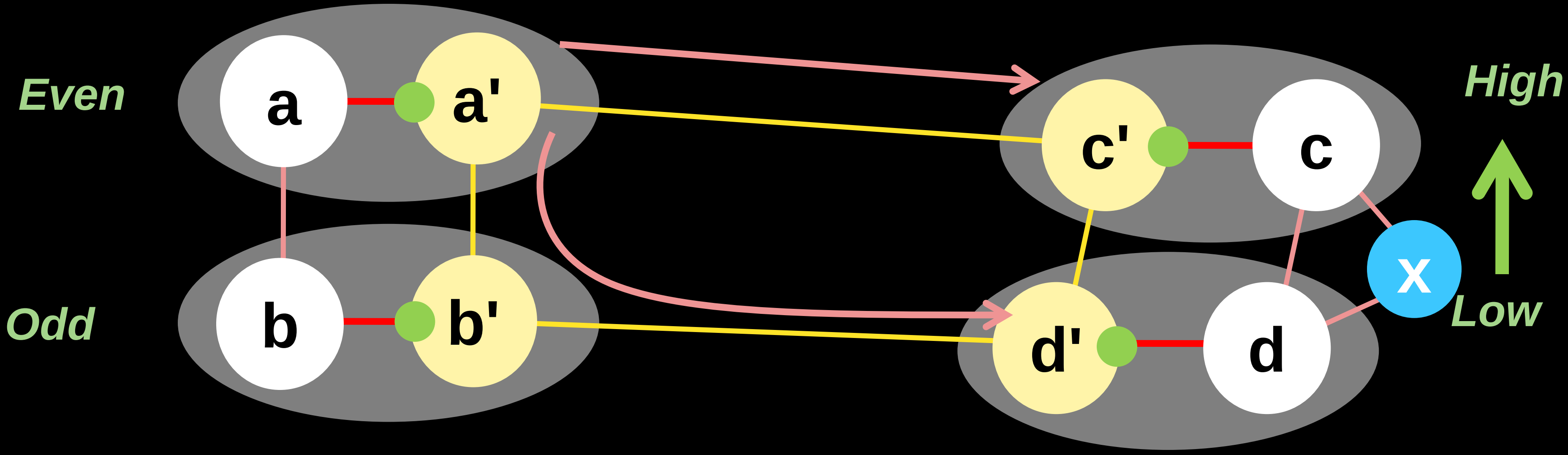
- When changing Gateway links, (here, moving the Even-Low Bundle from Gateway c'-c to d'-d) the situation is more complex.
- In the right-to-left direction, no handshake is needed. Node a' shuts off a'-b'-d' before turning on a'-c', whether using routing or protection switching.

Frame ordering: Gateway to Gateway



- In the left-to-right direction, a handshake is required between c' and d' to ensure that d'-d is turned off before turning on c'-c.
- Again, this is true whether routing or protection switching is used.

Frame ordering: Gateway to Gateway



- There is the further problem that, even if d'-d is closed before c'-c, congestion in the queues in c can cause node **x** to receive frames out of order.
- Fixing this requires that Gateway choice changes tie into the fault recovery protocol used in the Areas. This is a per-protocol issue.

Routing vs. Protection Switching

An interesting discovery (to be verified)

- As long as only one failure or recovery event occurs at a time, neither routing nor protection switching need handshakes to prevent forwarding loops – “break before make” solves all problems. **Both need handshakes** to prevent out-of-order delivery.
- If multiple events (a Gateway change and an inter-portal path change) take place simultaneously:
  1. **Routing** requires a handshake to prevent **forwarding loops** (or out-of-order delivery).
  2. **Protection switching** requires a handshake only to prevent **out-of-order delivery**; forwarding loops are prevented by the tunnel markers.

Area protection protocol support

What do Area Protocols need?

- As we know from our experience with MSTP and with various forms of L2GP (Layer 2 Gateway Protocol), making the decision as to which Gateway is to be used by each Service is not trivial.
- Ensuring against temporary loops or duplicate or out-of-order delivery when a change in this choice is made is even more difficult.
- It may be useful to the Area for the Buffer Network to provide a control path from one node of a Portal to the other nodes of the same Portal. For example, passing BPDUs would enable an MSTP Area to make safe Service-Gateway choices.

Control protocol requirements

Information passed in control protocol

- In order to switch frames in a manner that meets all of our goals, the following information must be distributed throughout the network:
- The state of every link, including the Gateway links.
- The preferences (demands) for which Gateway in a Portal each service is to pass through.
- Inter-Portal handshakes to ensure against temporary forwarding loops, if necessary.
- Administrator-optional Inter-Portal handshakes to ensure against out-of-order delivery.
- (Perhaps) a control path for the Area protocol to pass PDUs through the interconnect.
- Other items may be required by an existing protocol that we modify to suit this purpose.

Summary

Summary

- Whether we use bridging/routing technology, protection switching technology, or both, the **logical topology and the data flows are the same**.
- Protection switching has some advantages over bridging/routing because it requires less handshaking and thus can converge faster, but S-tagged services have tunnel identification issues.
- There are many possibilities for the protocol shared between the two providers if bridging/routing technology is used. We can enhance LACP, MSTP, CFM, or SPB, or we can invent something new.
- Enhanced CFM (ITU-T Protection Switching) is probably best for protection switching technology.

cisco