Failure rates and P802.1CB

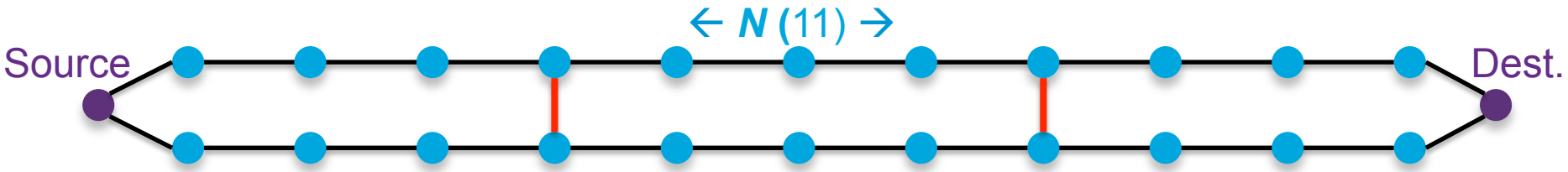
Norman Finn

Rev 1

October 29, 2013

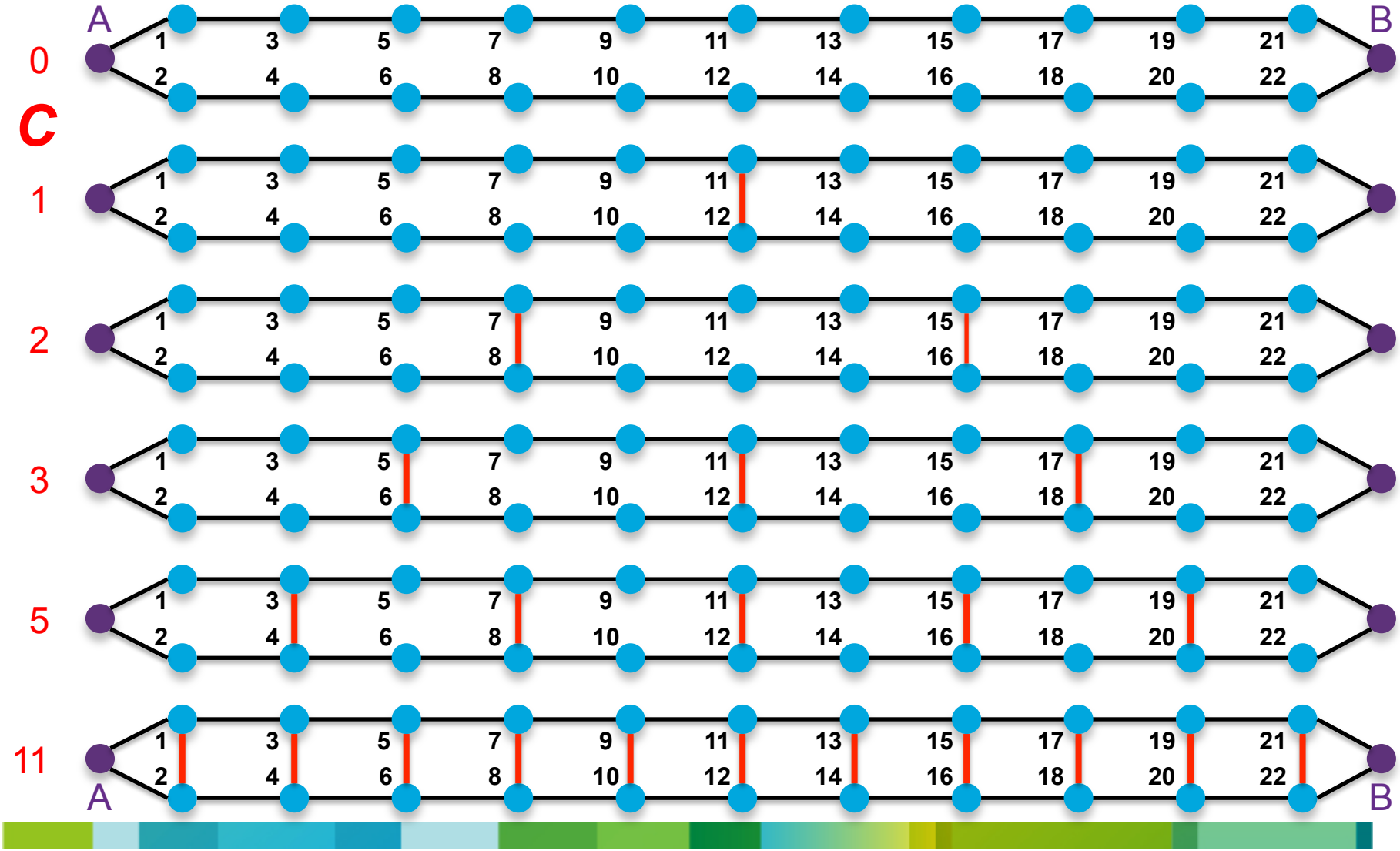
Method

Introduction

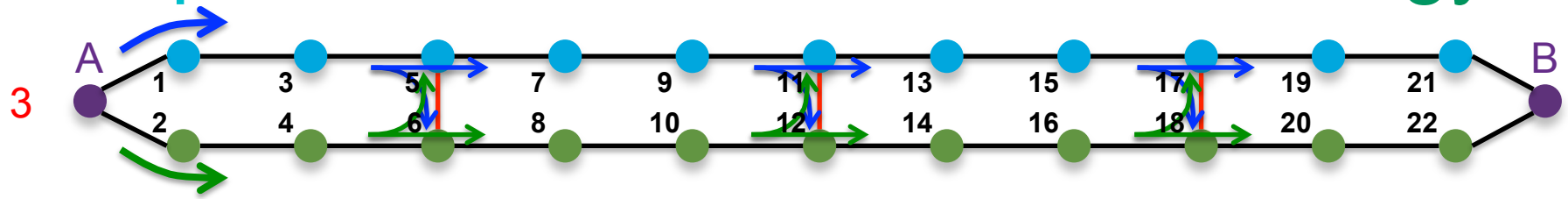


- The network topology analyzed in this presentation a ring with 2(*N*+1) **nodes**, two of which are the **source** and **destination** nodes. This creates two paths of *N* nodes each between the source and destination.
- The number of "**cross links**" between the two parallel paths from the source to the destination is varied, and the packet loss rates for different number of cross links is investigated.
- This presentation attempts to determine exactly how much the proposals for IEEE P802.1CB will improve the packet loss rate over the use of two or three paths without cross links.

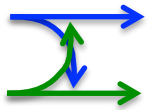
Rings with 24 nodes and *c* cross links



Dual paths with cross links – terminology



- Ring has **2*N*+2 nodes**, where ***N*** = 11 in this case.
- Source **A** and destination **B** never fail.
- Two paths, the **odd** and the **even** nodes, carry **duplicate data**.
- We will consider only **node** failures, not link failures.
- ***c* cross links** (*c* = 3 in the above example) allow paths to **crisscross**. Packets are replicated and/or discarded, thus ensuring data arrival even if one node in a main path fails.

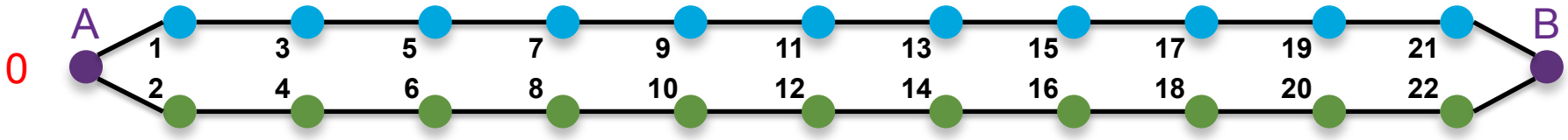


- Nodes attached to cross links (5, 6, 11, 12, 17, and 18) are replication/deletion ("**rep**") nodes and the rest are **non-rep** nodes.

Calculation method

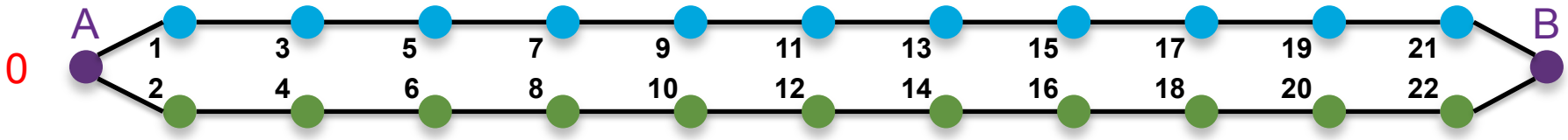
- If you ignore 3-node errors and assume that $(1–x)^y = 1 – yx$ (approximations that are within a few percent until you get to more than 100 nodes), then you can calculate the frame loss rate for two errors in a given network (P_{net2}) with *m* "bad" failure pairs that will cause a data loss in that network and a probability of a single failure P_F by:
  $P_{net2} = 1 – (1 – P_F^2)^m$
- That is, P_F^2 is the probability of a two-node failure, $(1 – P_F^2)$ is the probability of **not** getting a two-node failure, and $(1 – P_F^2)^m$ the probability of not getting *m* "bad" error pairs. One minus that is the probability of getting one bad pair, and thus is the network failure rate.
- So, the procedure for any network analyzed in this presentation is to calculate the number of "bad" pairs *m*, and use it in the above formula.

Analyzing 0 cross links



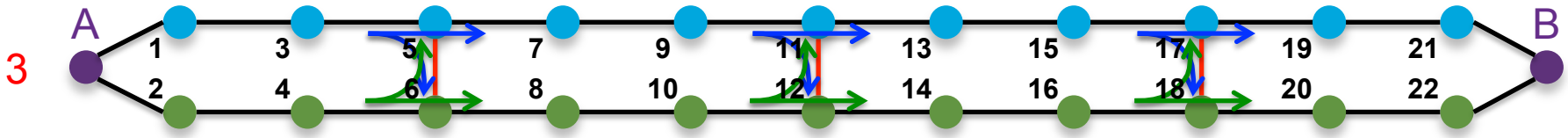
- There are 2***N*** nodes that can fail. Each has Mean Time To Failure (T_F) = either 10,000 hours or 250,000 hours, and Mean Time To Repair (T_R) = either 1 hour or 24 hours.
- The probability of loss P_F of a packet due to a node failure is $T_R/(T_F+T_R)$ = 1.0e-4 for {10000, 1}, 2.4e-4 {10000, 24}, 4.0e-6 {250000, 1}, or 9.6e-5 {250000, 24}.
- The packet loss probability for each *N*-node path is $P_{FN} = 1 – (1 – P_F)^N = P_{F11}$ (for $N = 11$)
  1.1e-3 for {10000, 1}, 2.6e-2 {10000, 24}, 4.4e-5 {250000, 1}, or 1.1e-3 {250000, 24}.

Analyzing 0 cross links



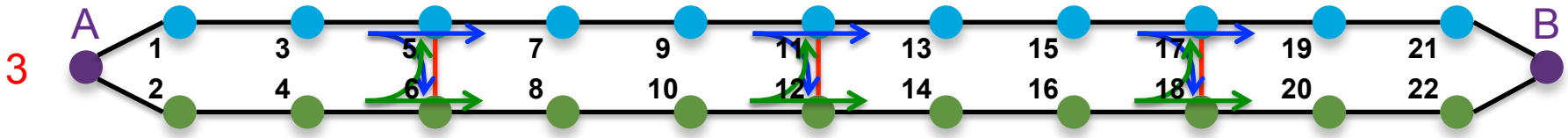
- The packet loss rate for a single failure is 0, since the other path still works.
- The probability of both paths failing is the square of the two paths' failure probabilities, or $P_{FN}^2 = P_{F11}^2 =$ (in this example) 1.2e-6 {10000, 1}, 6.8e-4 {10000, 24}, 1.9e-9 {250000, 1}, and 1.1e-6 {250000, 24}.
- (This is an exact calculation – it does not utilize the approximations.)

Analyzing *c* cross links



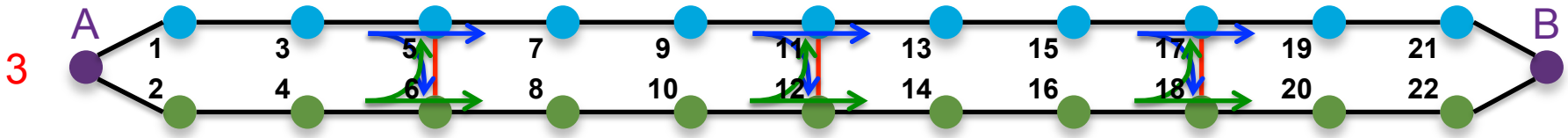
- We divide all pairs of nodes into **8 categories**. Each category is either "OK", meaning that this category of pairs does not stop both data streams, or "bad", meaning that it does stop both streams. We need a formula to count each class of pairs for *c* and *N*.
- **1. Same path, OK:** Some pairs of failures are in the same path, and have no effect (e.g. 1–3, 8–16, 7–21, above). There are 2 paths times *N*! / (2!(*N*–2)!) pairs per path = 110 such pairs for *N* = 11 and any value of *c*.
- **2. Rep/rep pair, bad:** Some failures involve directly-connected pairs of rep nodes, and cause a delivery failure (5–6, 11–12, 17–18). There are *c* of these, or 3 pairs for *c* = 3 and any value of *N*.

Analyzing *c* cross links



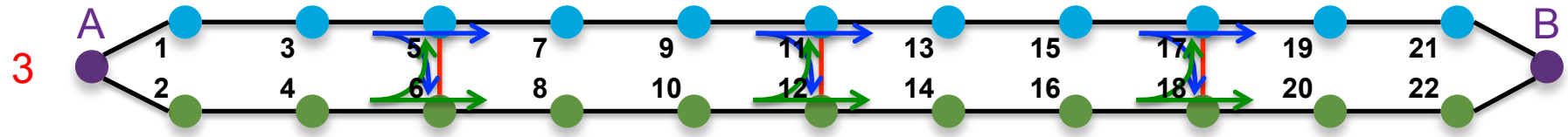
- **3. Non-rep/non-rep, bad:** If a pair of non-rep nodes in the same "cell" (between the same pair of replicators) fail, then both paths are broken (e.g. 1–4, 9–8 above, but not 7–14). There are (number of nodes per cell)^2 * (number of cells) of these = $((N - c) / (c+1))^2 (c + 1) = 16$ pairs for $c = 3$, $N = 11$.
- **4. Non-rep/non-rep, OK:** Some pairs of failures are in two different paths, but do not affect rep nodes, and so do not affect data delivery (e.g. 3–20, 13–16 above, but not 19–20). There are (number of nodes per cell between cross links)^2 * ((number of cells) * (number of cells – 1)) such pairs = $((N - c) / (c+1))^2 (c + 1)c = 64$ pairs for $c = 3$, $N = 11$.

Analyzing *c* cross links



- **5. Rep/non-rep, bad:** Some failures involve a rep node and a nearby non-rep node, and cause failures (e.g. 5–4, 5–10 above, but not 5–14). There are 2 paths * (2 * (number of non-rep nodes per cell)) * (number of rep pairs) = 4 (*N*–c)/(*c*+1)*c* = 27 for *c* = 3, *N* = 11.
- **6. Rep/non-rep, OK:** Just like the last class, but the non-rep failure is far enough away to not cause complete data loss (e.g. 5–14, 11–2 above, but not 11–10). There are 2 paths * ((number non-rep nodes per path) – nearby non-rep nodes) * (rep nodes per path) = 2 (*N* – *c* – 2(*N*–*c*)/(*c*+1))*c* = 36 pairs for *c* = 3, *N* = 11.

Analyzing *c* cross links



- **7. Rep/rep nearby, bad:** When one rep node from each path fails, and the rep nodes are in adjacent pairs, the result is frame loss (e.g. 5–12, 17–12 above, but not 17–6 or 17–18). There are 2 paths * (number of adjacent pairs of rep pairs) of these = 2 (c – 1) = 4 of these pairs for c = 3 and any value of N.
- **8. Rep/rep far, OK:** When one rep node from each path fails, and the rep nodes are not in adjacent pairs, the result is benign (e.g. 17–6, 5–18 above, but not 5–12 or 17–12). The number of such paths is the number of possible rep/rep pairs – the number of adjacent pairs – the number of nearby pairs = $c^2 – c – 2(c–1)$ = of these pairs for c = 3 and any value of N.

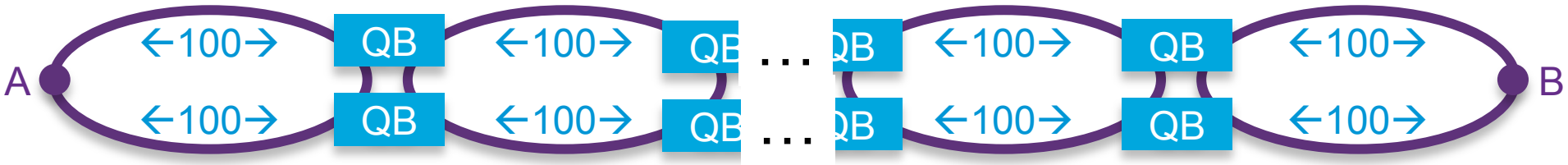
Results

Results

- A spreadsheet (also uploaded) contains the calculations.
- Inaccuracies of a few percent are expected for smaller *N*, and may be worse for *N* = 1001.
- Various assumptions about MTBF and MTTR were calculated; two combinations are presented on the next page.
- Various sized networks (11, 107, and 1001 nodes) and various numbers of cross links (0–3, 5, 11, 107, 314, 10001) were used.
- The right-hand five columns show the the exact loss rate if no cross links are used, the calculated (approximate) loss rate using the cross links as has been suggested for P802.1CB, the ratio of these two (ratio > 1.0 == P802.1CB improves reliabiity), followed by the same two columns for three paths with no cross links.

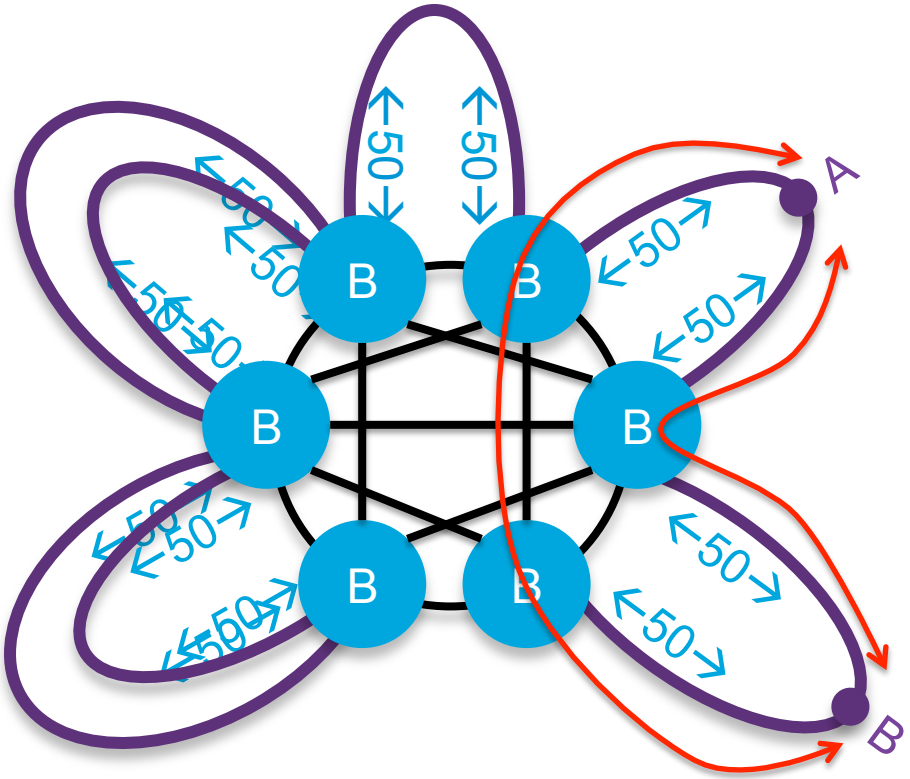
Nodes/path (2 paths in network)	Number of P802.1CB cross links	MTTF (hrs)	MTTR (hrs)	Exact loss rate for 2 paths, no cross links	Approximate loss rate using cross links	Improvement factor using cross links	Exact loss rate for 3 paths no cross links	Improvement factor using 3 paths
11	0	10,000	24	6.80e-04	6.97e-04	0.98	1.78e-05	38
11	0	250,000	1	1.94e-09	1.94e-09	1.00	8.52e-14	22728
11	1	10,000	24	6.80e-04	4.09e-04	1.66	1.78e-05	38
11	1	250,000	1	1.94e-09	1.14e-09	1.70	8.52e-14	22728
11	2	10,000	24	6.80e-04	3.17e-04	2.15	1.78e-05	38
11	2	250,000	1	1.94e-09	8.80e-10	2.20	8.52e-14	22728
11	3	10,000	24	6.80e-04	2.71e-04	2.51	1.78e-05	38
11	3	250,000	1	1.94e-09	7.52e-10	2.57	8.52e-14	22728
11	5	10,000	24	6.80e-04	2.25e-04	3.03	1.78e-05	38
11	5	250,000	1	1.94e-09	6.24e-10	3.10	8.52e-14	22728
11	11	10,000	24	6.80e-04	1.79e-04	3.81	1.78e-05	38
11	11	250,000	1	1.94e-09	4.96e-10	3.90	8.52e-14	22728
107	0	250,000	1	1.83e-07	1.83e-07	1.00	7.84e-11	2337
107	1	250,000	1	1.83e-07	9.33e-08	1.96	7.84e-11	2337
107	2	250,000	1	1.83e-07	6.33e-08	2.89	7.84e-11	23367
107	3	250,000	1	1.83e-07	4.84e-08	3.79	7.84e-11	2337
107	5	250,000	1	1.83e-07	3.34e-08	5.48	7.84e-11	2337
107	11	250,000	1	1.83e-07	1.84e-08	9.94	7.84e-11	2337
107	107	250,000	1	1.83e-07	5.10e-09	35.88	7.84e-11	2337
1001	0	250,000	1	1.60e-05	1.60e-05	1.00	6.38e-08	250
1001	1	250,000	1	1.60e-05	8.03e-06	1.99	6.38e-08	250
1001	2	250,000	1	1.60e-05	5.37e-06	2.98	6.38e-08	250
1001	3	250,000	1	1.60e-05	4.03e-06	3.96	6.38e-08	250
1001	5	250,000	1	1.60e-05	2.70e-06	5.92	6.38e-08	250
1001	11	250,000	1	1.60e-05	1.37e-06	11.69	6.38e-08	250
1001	107	250,000	1	1.60e-05	1.80e-07	88.62	6.38e-08	250
1001	314	250,000	1	1.60e-05	8.28e-08	192.75	6.38e-08	250
1001	1001	250,000	1	1.60e-05	4.80e-08	332.56	6.38e-08	250

Where does P802.1CB **help**?



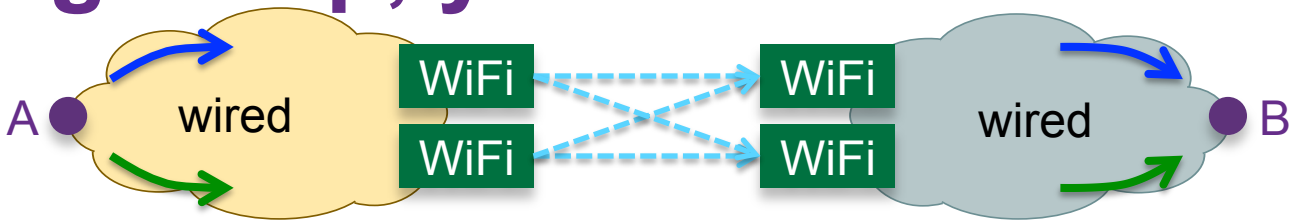
- **IF** one is limited to using only ring topologies with dual connections between rings (as in HSR); and
- **IF** one has a very large network (2000 nodes, above); and
- **IF** one must arrange the rings linearly;
- **THEN** using something like 802.1CB at the ring interconnects can improve reliability by a factor of 10 or so. (See the last few lines of the chart.)
- **BUT** that is a lot of effort for not much gain.

Why bother with chains of rings?



- **IF** one is **not** limited to using only ring topologies;
- **THEN** other topologies make much more sense, and give better reliability than the chain of rings, without P802.1CB.

Don’t give up, yet!



- **IF** one has a particularly unreliable medium, such as Wi-Fi; and
- **IF** one replicates a number of packets separated in time, frequency, and space (8 on 4 links in the example above);
- **THEN** using something like 802.1CB at the wired/wireless interconnects may significantly improve the reliability of the unreliable connection between two clouds.
- **BUT** this scenario has not been thoroughly investigated (at least, by this author).

Tentative conclusions by the author

- Adding *c* cross links improves reliability by a factor of about (*c* + 1), because it reduces the proportion of two-node failures that stop data.
- As you add cross links, reliability improves slowly, because until the number of cross links approaches the path length, there are still lots of failure pairs that stop the data.
- Note also that:
  1. Every cross link carries twice the traffic of a normal bi-directional link, so is either a limitation to the network bandwidth, or requires a two-link aggregation;
  2. Thus, you may have to double the number of ports in the network to get a complete full-bandwidth ladder to improve the reliability by a significant amount.
  3. The technique proposed to date requires a tag header and hardware state in every replicator/discarder in the network, and requires more complexity to work with streaming data (packet rate > difference in delivery time over the paths).
- **P802.1CB may have limited applicability, and low value for the effort. Let us investigate, further. Perhaps there are good uses.**

Further work

Readers are encouraged to extend this work in order to test whether the tentative conclusions are valid. Among the possibilities:

- Correct any gross blunders in this presentation.
- Analyze other topologies besides the partial ladder.
- Consider not just node failures, but link failures, especially unreliable links.
- Perform more accurate calculations of the probabilities, avoiding the approximations made here.

Thank you.