



# New Uses for Synchronization in Network Infrastructure



**4th International Telecommunications Synchronization Forum  
November 15th, 2006**

**Stewart Bryant**  
**stbryant@cisco.com**

# Time in networks

- Traditional packet switch networks have been designed without the use of any form of sophisticated timer support.
- The protocols that are used to create and support the network topology rely entirely on message synchronisation with coarse timers as a backstop.
- Network Time Protocol is available but still only provides a relatively coarse time accuracy in the range 500us in a constrained network to 50ms+ in a general purpose network.

# Time in Packet Networks

- A number of new applications for packet networks are emerging that are time sensitive.
- This is leading to the introduction of high quality synchronization services.
- However as we will explore, the availability of high quality synchronization leads to new knowledge of network behaviour, and to new methods that may be applied to the design of distributed systems.

# The Beachhead Applications

The provision of high quality synchronisation in packet network infrastructure is being driven by the need to provide precision clock to two types of client network infrastructure:

1. Lower layer synchronisation beyond the packet network, for example mobile phone base stations in support of their radio systems.
2. Circuit emulation over packet: TDM, SONET/SDH, ATM etc, using mechanisms standardised by the IETF PWE3 WG and its colleague groups in other SDOs.

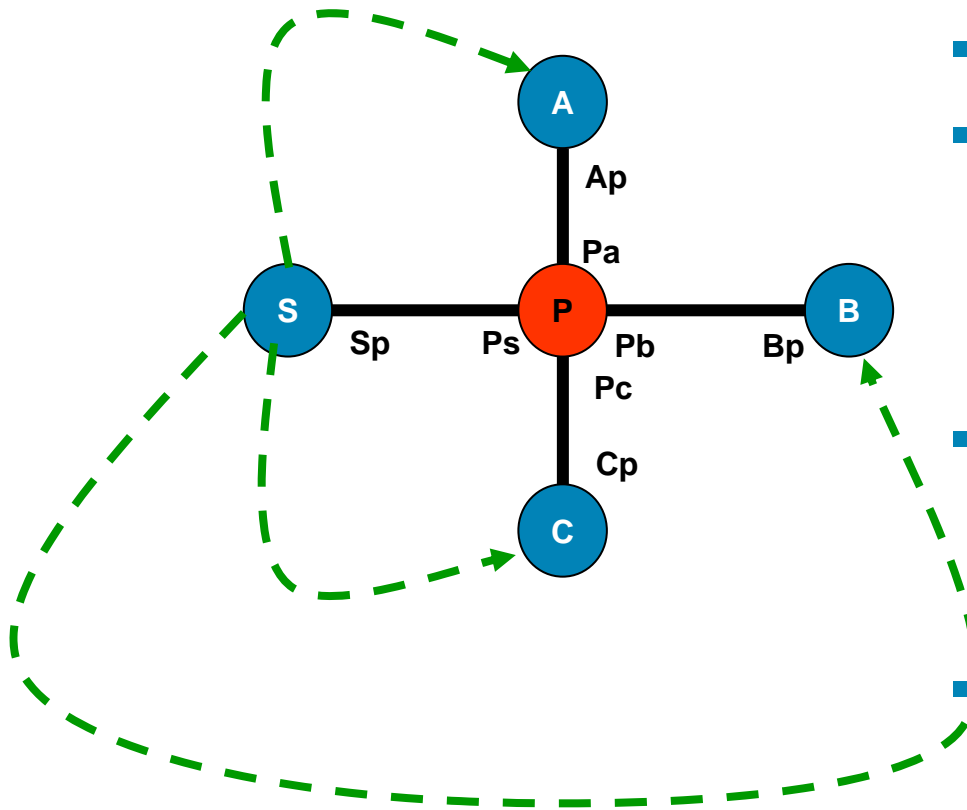
# Packet Time of Flight

- The new service types that are being operated over packet networks are more delay and jitter sensitive than was historically the case.
- Network structures are becoming virtualised with multiple layers of encapsulation that hide the true path of packets.
- Network bandwidth usage is asymmetric in terms of traffic, bandwidth and path.
- These trends lead to a requirement for one way packet delay measurements with more precision than we have used in the past.
- Packet time of flight can only be measured between points in the network that are time synchronized.

# Network Dynamics

- The demand for higher network up times has led to a demand for faster network convergence following failure. The multi-second convergence times of the past are being chased down to convergence times that approach 200ms.
- In cases where 200ms is inadequate, fast reroute techniques are being deployed with a reroute time of better than 50ms.
- To execute a repair in better than 50ms it is necessary to detect and confirm the failure in significantly less than 50ms.
- To monitor and diagnose the topology and packet flow during these events it is necessary to provide packet forensics with a time resolution that is significantly better than is currently the case.
- To study the dynamics of a network during a topology transition, time synchronisation of the routers is needed.

# Fast re-route



- P fails.
- No routers other than the neighbours of P are aware of the failure and there is no time to tell them and no time for them to react.
- The neighbours of P use a predetermined repair plan (frequently a tunnel) to bypass the failure.
- The network then enters a controlled convergence phase to restore native operation in the new topology (more about that later).

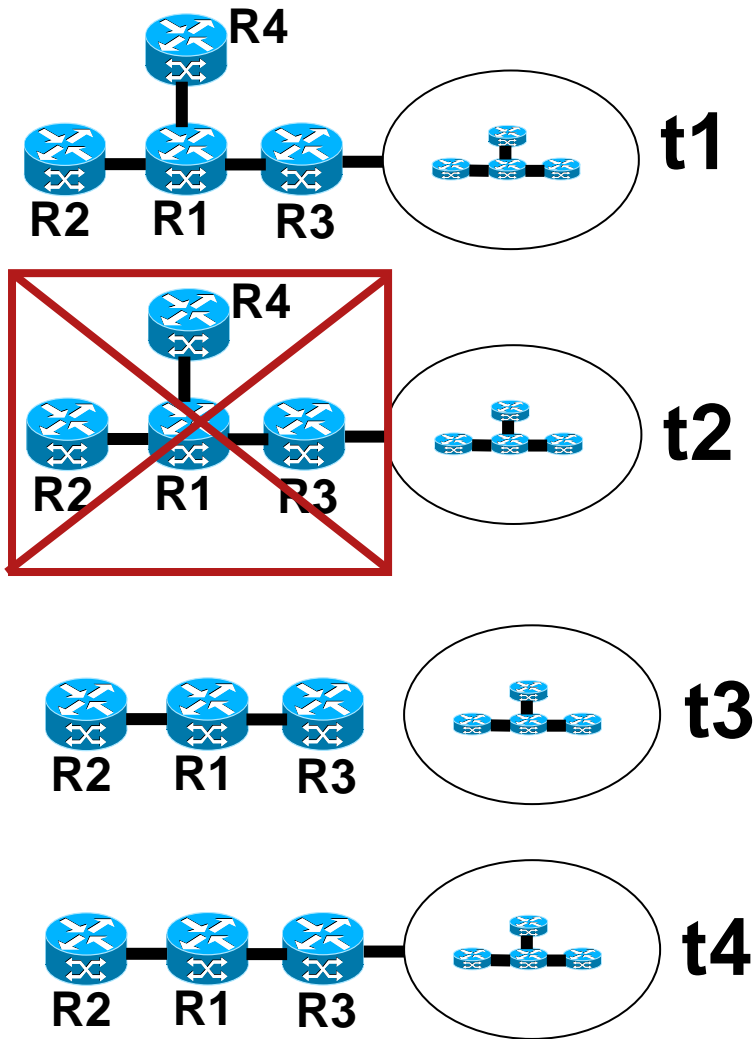
# Protocol Design

- Classical distributed systems have been designed on the basis that a common view of time is not available.
- Synchronisation is therefore through messaging with coarse timers used to initiate exception processing to catch cases where messages are lost.
- This leads to complex synchronisation algorithms.
- Distributed systems must be self stabilising in the event of ANY error.
- The Internet crashed only once (when it was young). The homogeneous nature of the routers meant that it could be restarted without a complete (manual) shutdown of all routers.
- Such a repair would not be possible today.
- The reason was related to an earlier design of the link state flooding mechanism.

# Example 1 – Link State Packet (LSP) Confusion

- Most IP networks exist in a state of dynamic stability in which they discover the available routers automatically and use a dynamic routing protocol to determine the topology and optimum (shortest) paths to destinations.
- One class of routing protocol is called Link State Routing (e.g. ISIS and OSPF). The basic mode of operation is that every router tells all other routers who its neighbors are. Based on this information the shortest path is calculated based on Dijkstra's algorithm.
- For this to work all routers must have a complete and congruent view of the network link states.
- If the routers do not have a complete and congruent link state database, the result is black holes and packet loops.

# Example 1 – LSP Confusion



- Consider the case of group of routers R1 ... R4 attached to a network.
- R1..R4 crash (power failure).
- R1, R2 and R3 return to service and converge but are not yet attached to the network.
- The network has not yet timed out R1..R4 link state information and still thinks R4 exists.
- R1..R3 know otherwise, but in some circumstances the contradictory information has the same sequence number when they reattach to the network (the proxy used for time)
- This contradiction will exist until it is resolved by the routing protocol.

# Example 1 – LSP Lifetime Confusion

- When a router detects that it has two LSPs with the same sequence number, but different checksums, it treats it as if the LSP lifetime has expired. This flushes the information from the network which causes the originator to re-flood the up to date information.
- The protocol thus recovers.

# Example 1 – LSP Lifetime Confusion

- If time had been a ubiquitous commodity in the network at the time when these protocols had been designed (late 1980s), the link state packet would have simply stated its time of origin and there would have been no ambiguity.
- There are a number of more important cases where time would have simplified the protocol, for example LSP lifetime expiry synchronisation. However, they are more complex to explain.
- Time therefore would have simplified the protocol design.
- However there would have been the recursion problem.

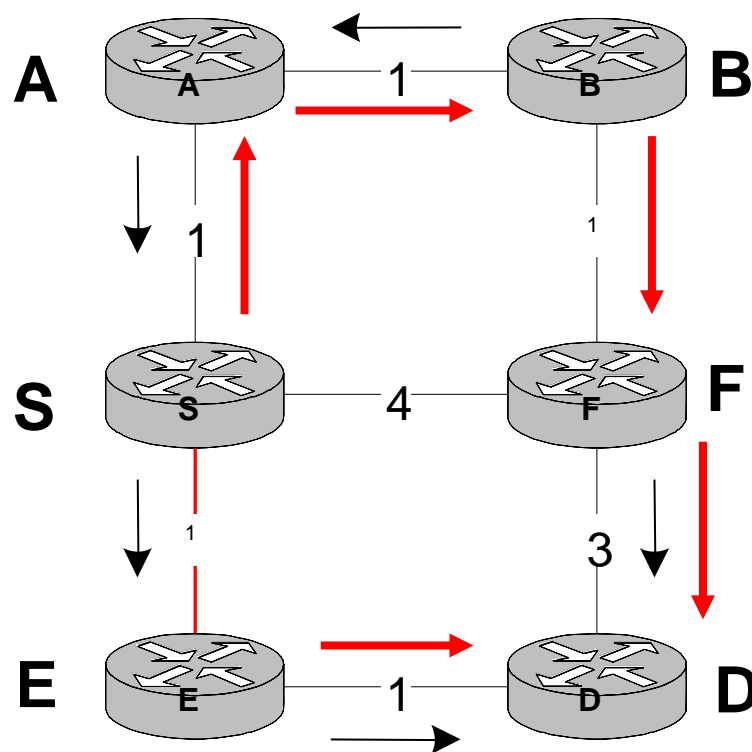
## Example 2 - Micro-loops Are Bad

- When the topology of a network changes it needs to re-converge on the new topology.
- This requires routers to update their forwarding tables.
- They normally do this by racing to do this in the shortest time possible.
- During this time there may be inconsistency between adjacent routers, and if the result is mutual next hops then micro-loops form.
- Micro-loops result in collateral damage to traffic not affected by the change, as well as causing the affected traffic to be lost.

# Example 2 - What Are Micro-loops?

Paths to D AFTER  
Failure of S-E

Paths to D BEFORE  
Failure of S-E



If A changes before  
B, we have a loop  
across A-B  
until B changes

Note that we have a potential loop *anywhere* that the red and black arrows are in different directions.

## Example 2 – Micro-loop Strategies

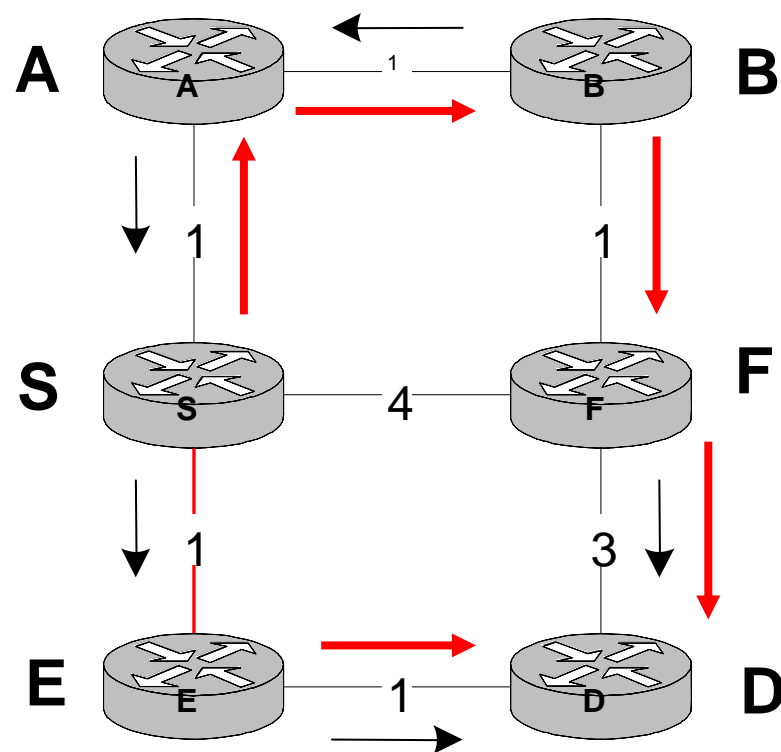
- There are a number of micro-loop mitigation and prevention strategies.
- They are either slow, require tunnelling (virtual path) hardware, or require the use of non-trivial protocol design.
- Consider ordered FIB\* (one of my two favourite approaches).

\*FIB = Forwarding Information Base

# Example 2 - Ordered FIB Mechanisms

- Change AB cost incrementally
  - Bleeds traffic away from AB one layer at a time.
  - Simple, but slow
- Calculate order using reverse SPF routed at far side of failure.
  - Faster, but takes time
- Signal that FIB change is complete
  - Fast, but signal packets must never be lost
- Combination of Ordered FIB and Signalling
  - Calculate order and set change timer
  - Change early when all children say they are done
  - A good compromise between speed and resiliency

## Change FIBs in order BAS



# Example 2 - Synchronized FIB Swap

- Routers Signal/determine time to change.
- Second FIB prepared with new topology.
- Routers change FIB at predetermined time.
  
- There are hardware resources issues with this approach.
- It is only as good as the time synchronisation.
  
- BUT it is dramatically simpler than any of the alternatives.

# The Recursion Dilemma - Revisited

- You can only rely on time if it is provided by a client layer.
- That lower layer may be one of your own sub-layers.
- However provided you design a simple backstop mechanism you can use time at the same layer to make the network work better.

# Conclusion

- Packet networks have traditionally operated without the availability of a precision time service.
- New applications require the introduction of a time synchronisation service.
- That time service will give us a better insight into the dynamic behaviour of the network and the traffic carried by it.
- That time service will also us to consider new protocol mechanisms, both within the network itself and at the network application layers.

# Q and A



