Synchronisation of Objects in a Broadcast Network

Dr J C Wrigley, STASYS Ltd.

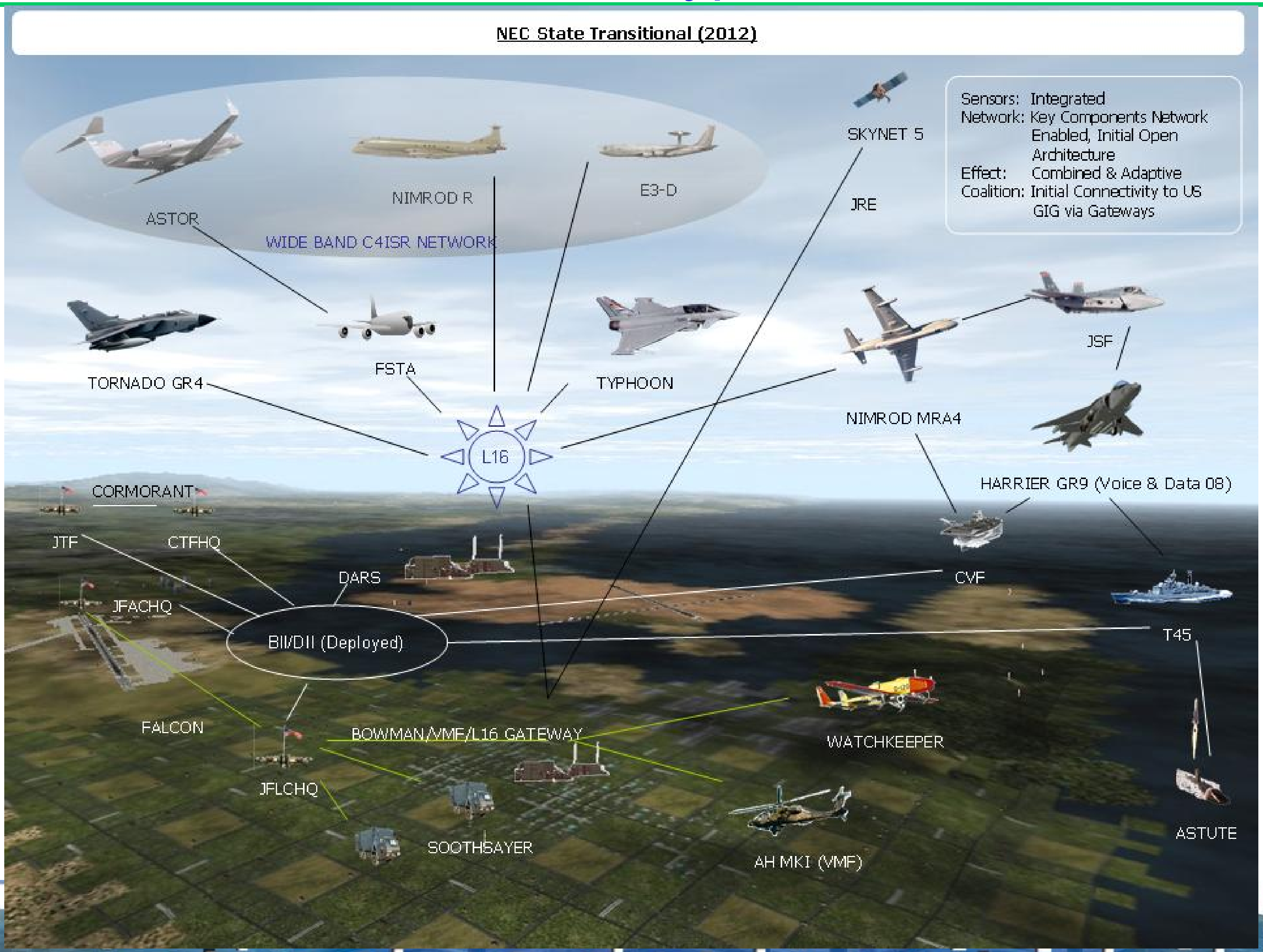
About this presentation

- My background (20yrs) is in Tactical Datalinks (TDLs)
- However, recognise that TDLs are the end of one particular road
- Need to apply newer technologies
- Will brief a concept. Implementation requires better XML engineers than me!

Why is this of interest ?

- Military applications require a near-real time picture of the tactical environment, formed from multple networked sensors.
- Typical solution to date is to use a military datalink, a messaging system with a broadcast carrier.
- Tactical environment uses mobile radio infrastructure, currently low-medium bandwidth (kbps)

Typical datalink scenario



The basic problem

- “Maintain a common, synchronous picture between all tactical units.”

- Engineering version: “maintain synchronised collection of objects between multiple platforms, using a low bandwidth broadcast network”

TDL solution

- TDLs originally derived from teletype messaging systems (eg Link 14)
- 60s (approx) development of Link 11, HF/UHF roll call system
- Uses fixed format messages representing tracks
- “Database” implementation left to host system

TDL history

- Then in 70s to date, development of Link 16
- Link 16 messages based on Link 11 messages – message centric design
- TDMA UHF carrier (MIDS), plus now SATCOM options
- Database and protocol implementation still a host issue

Problems with the TDL approach

- Message Centric
  - Host database and protocol implementations still platform-specific, leads to interoperability problems
  - Some mitigation of this by the use of iSMART process – a clearer definition of database structure and protocols

- Hand crafting and CM of messages
  - Is a major industry. Causes long time lags in the implementation of new operational capabilities. No guarantees of optimal message design.

Time to rethink the problem

- Now the 21st century- there must be a better way

  - Throw bandwidth at it. Not always possible on smaller and legacy platforms.

  - Analyse as a database synchronisation problem

Database synchronisation

- Typically considered for LAN/WAN systems – high integrity networks, good assurance of data transfer.
- Exchange of messages between databases, driven by changes to each database. Messages are generated automatically by the system
- Need to consider in context of tactical system – lower QoS, no need for exact synchronisation, latency trade-offs.

Statistical synchronisation

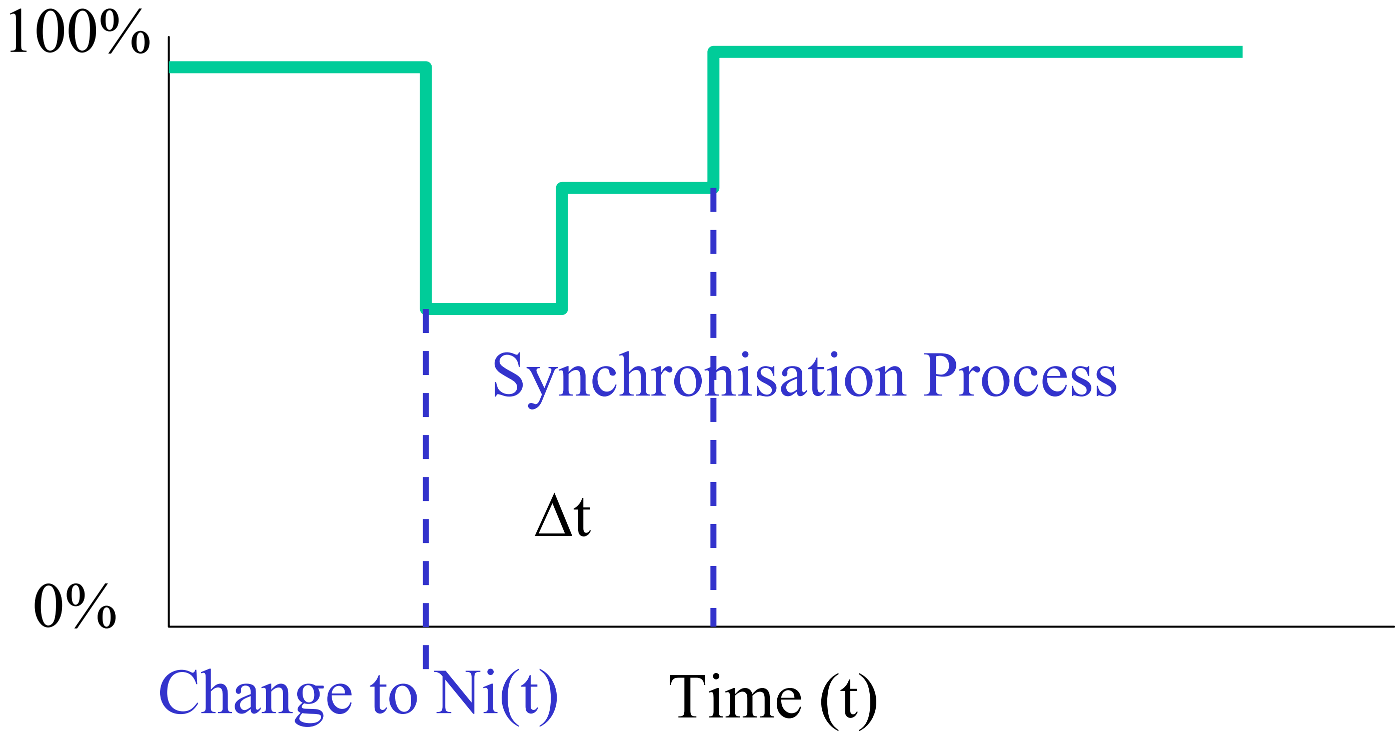
- In military environment, may (have to) tolerate incomplete synchronisation
  - Bandwidth not sufficient for assured delivery protocols
  - Some data time-expired anyway, so no sense in re-trying.

Measuring and bounding the errors

- Database/set of objects can be represented by a single (large) number (eg concatenate all records/linearised objects), say $N_i(t)$ for ith participant at time t.
- Problem is to ensure that $N_i(t) \approx N_k(t)$ for all i, k and t
- Limited by latency in system and QoS of bearers
- Practically can only ensure that $Diff(N_i(t), N_k(t)) \leq \Delta$

Graphically

- Suppose 100% represents equality and 0 complete lack of correlation between $N_i(t)$, $N_k(t)$:



Latency

- Δt, time lag, depends on system bandwidth

- Simple formula – if L bits need updating, then
  - $\Delta t = kL/B$, where B is bandwidth in bits/sec. (actually channel rate). k represents the efficiency of the messaging protocol

TDL application

- Example TDL : 20 participants, latency reqt 2s, tracks 50 bits, No of tracks requiring update: 100. (Note : not real numbers!)

- If done by pairwise exchanges need
    - 20 x 19/2 x 50 x 100/2 = 475 kbps

- If done by broadcast, need
    - 20 x 50 x 100/2 = 50 kbps

R2 protocol

- TDLs also use an R2 protocol to try to ensure that only one participant reports on each object. Bandwidth requirement down to:
  - 50 x 100/2 = 2.5 kbps
- Much more commensurate with battlefield carriers.
- Tradeoff is lack of assurance of commonality of databases – now only statistical and only one source per track – lose possibly useful data

How can we re-engineer within these constraints ?

- Go data-centric
  - Remove reliance on hand-crafted messages
  - Clearly define synchronisation protocols
  - Clearly define host system object model
- Consider constraints on tactical systems
  - Synchronisation can only be as good as bandwidth permits

Data centric solution

- Concentrate effort on defining model of tactical objects – eg tracks
- Define an automatic protocol for synchronising object collections between users
- Benefits
  - Users no longer involved with message construction and protocol issues
  - Easier to extend and add objects to system
  - Can use existing bearers
  - More bandwidth efficient (possibly)

Proposed Implementation Program

- 1) Define object model in XML
- 2) Define an exchange protocol to synchronise objects between platforms
  - SOAP
  - Use WS-CDL to define protocols ?
  - Define a packed binary serialisation process
- Map onto existing bearer PDUs (eg MIDS 70 bit words)

Example

- Example protocol:
- For each attribute of each tactical object, define the following meta-data:
  - Maximum latency (d)
  - Periodicity requirement (p)
  - Last periodic report time (tp)
  - Last data change time (td)
  - Assured delivery required
- Scan the objects for periodic updates and “on change” update requirements
- Construct messages to update affected attributes
- Broadcast messages
- Repeat until ack or fail if assured delivery required

But will this mean more bandwidth is required?

- Concern that potentially more smaller messages will be required, increasing overhead

Bandwidth efficiency

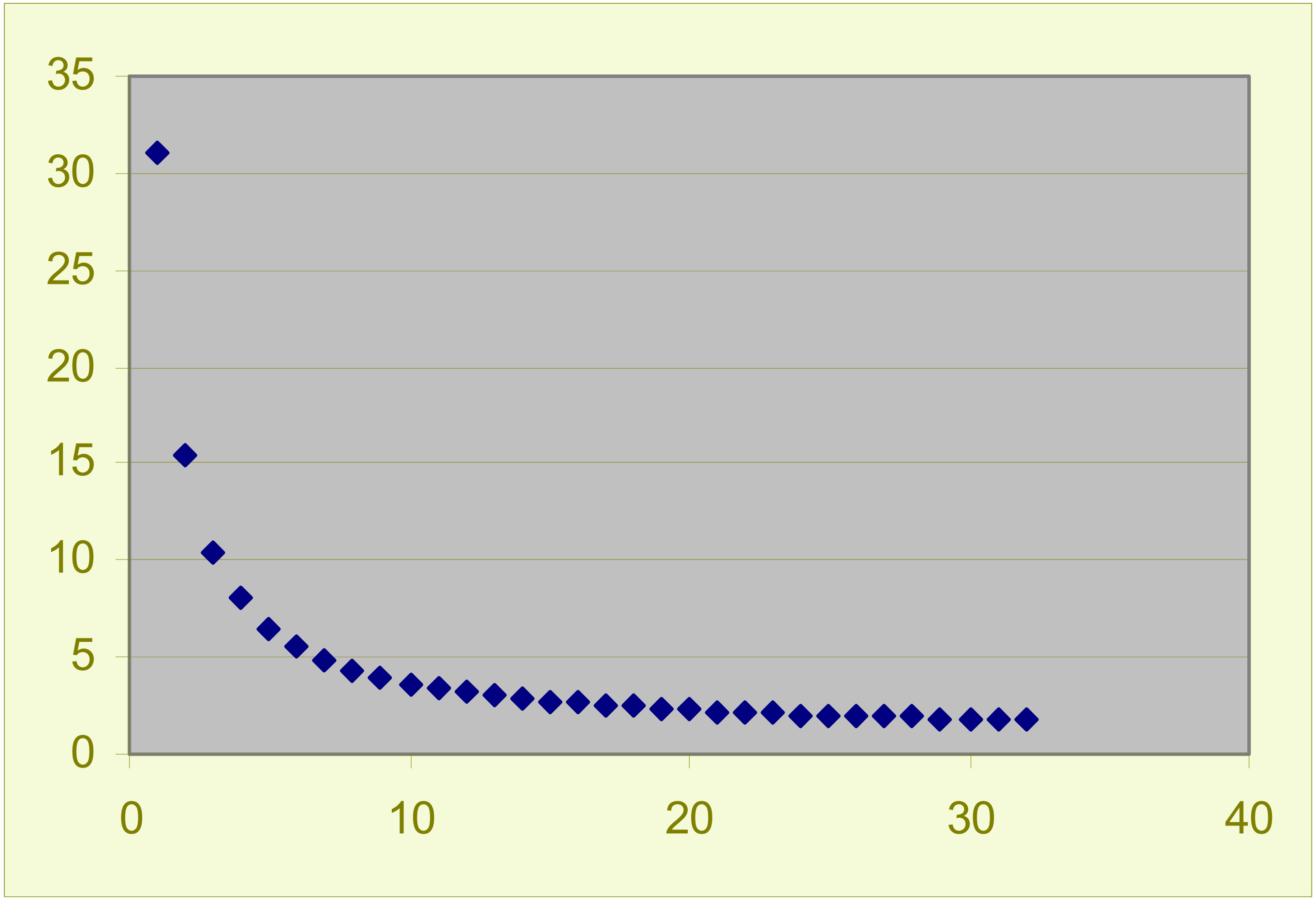
- Suppose database is N bits
- Define a protocol word as w bits
- Require $\log_2(N/w)$ bits to define segment to be updated (ie address)
- Eg – database size 4096 bits, word size 32 bits, index needs 128 values, ie 7 bits.

Bandwidth utilisation

- To update one segment – message length is:
  - $w + \log_2(N/w) + h$ (header length in bits)

- To update (say) a fraction f of the database, bandwidth requirement (broadcast system with R2) is:
  - $fN/w\ (w + h + \log_2(N/w))/\ \Delta t$

Example graph of bandwidth vs "word" length



(h = 16, N = 15,000)

Optimum word length

- Bandwidth ratio tends to 1 for large w
- Datalinks operate in w=100 region
  - Aside on TNs: TDLs use large indexes for these. Suggest handle as an attribute and use smaller db index
- Propose smaller words for automated mechanism – down to single attribute changes
- 32 bits gives acceptably low overhead and accommodates most track attributes (eg lat/long/TN)

Proposal

- Express tactical objects in XML
- Form PDUs by encoding attribute changes into 32-bit words using defined protocols
- More efficient than current TDLs – not all bits for an object transmitted every time

Conclusion

- Have developed a practical way to improve on current TDLs
  - Data centric – easier to add new object types
  - No CM of message standards. Instead CM of object model which is more closely linked to user view
  - At least as efficient as current TDLs
  - Can reuse same bearers
  - Host can be isolated from messaging implementation details

Work to be done

- 1) Develop object models (XML)
- 2) Develop protocols, binary representation and serialisation processes (SOAP +)
- 3) Replace TDLs

Implementation

- Can we avoid major costs in legacy systems ?
  - One-off cost of migrating to an object-oriented host interface
  - Underlying messaging system then transparent to host – can be Link 16, Link 11, XML binary, etc.