New Automotive gPTP Applications and Their Performance Needs

Don Pannell, Fellow
Automotive Ethernet Networking, NXP Semiconductor

Rajeev Roy, Senior Principal
Automotive Ethernet Networking, NXP Semiconductor

IEEE Ethernet & IP Tech Days – November 2022

SECURE CONNECTIONS
FOR A SMARTER WORLD

OVERVIEW

- Use Cases Requiring more Accurate gPTP
- Options to Improve gPTP's Accuracy
- Summary & Questions

Special thanks to our NXP colleagues:

Feike Jansen, Maik Brett and Stefan Singer,

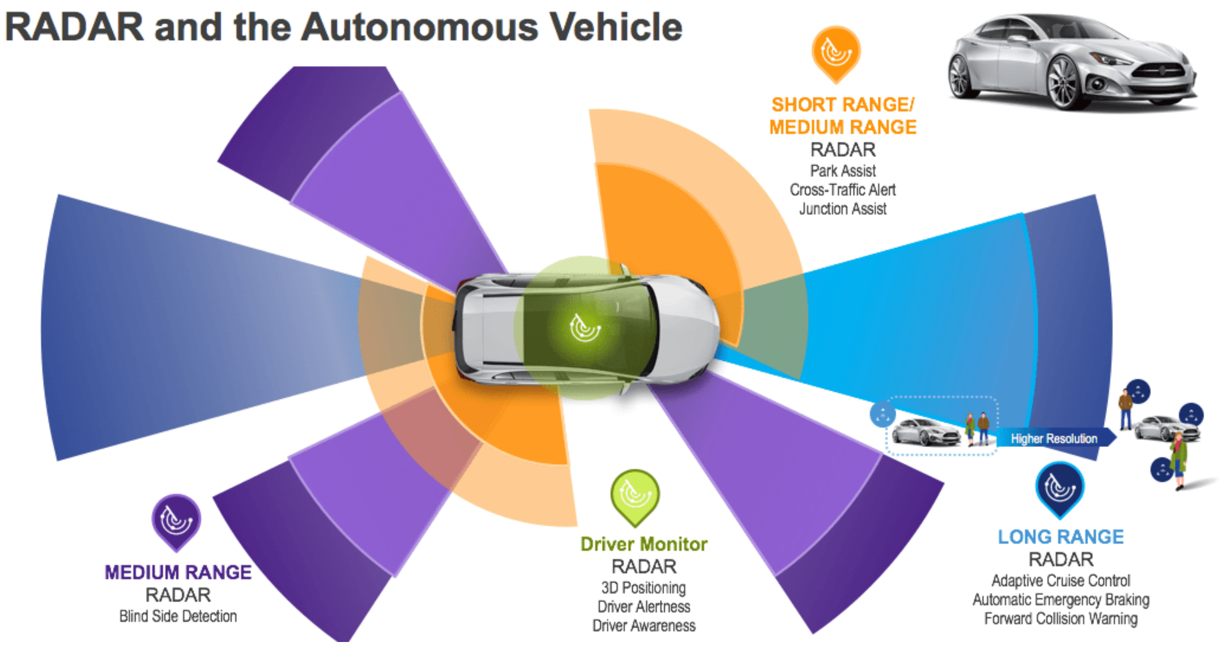
and David McCall, Intel, for his gPTP work

Use Cases Requiring More Accurate gPTP

The Time Sensitive Networking solutions

SECURE CONNECTIONS
FOR A SMARTER WORLD

RADAR IN AUTOMOTIVE

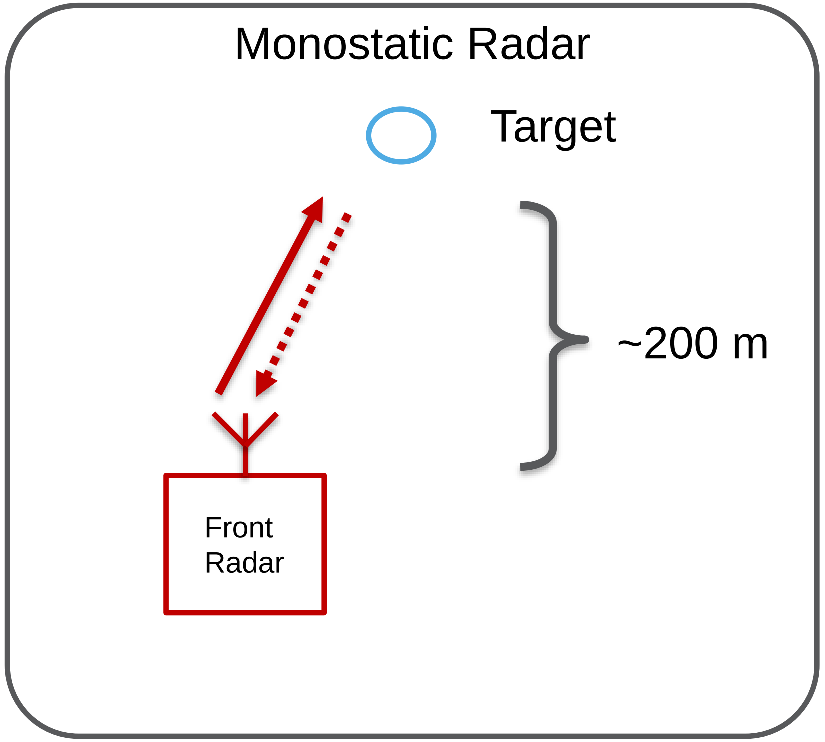


Type of Radar	Typical Range (m)
Short Range Radar (SRR)	0.15-0.75
Medium Range Radar (MRR)	0.3-150
Long Range Radar (LRR)	0.6-300

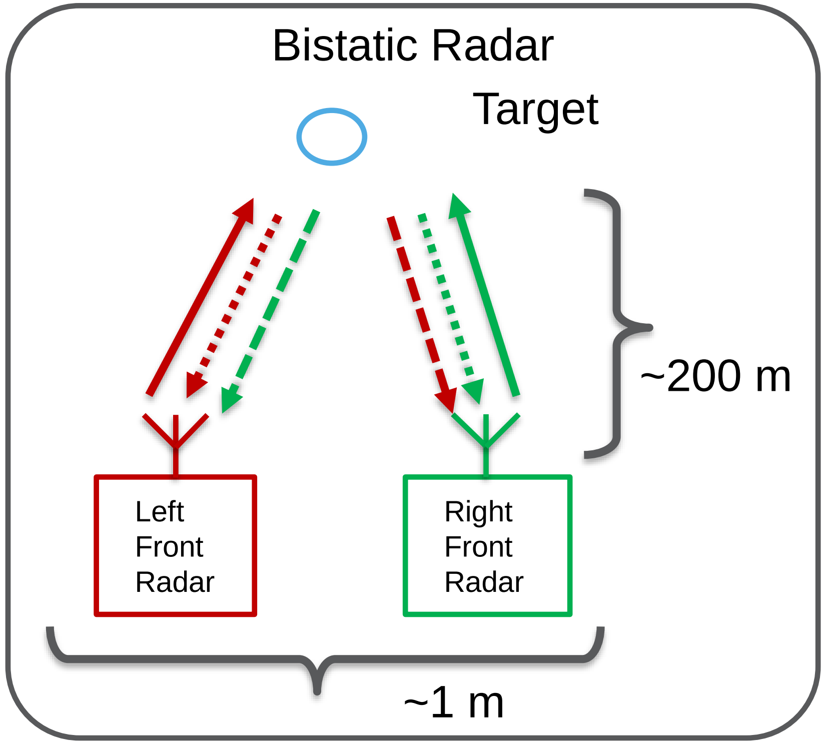
RADAR PRINCIPLES-1

Use of a Radar:

- Measurement of
- Distance
- Velocity
- Angle



Typical usage of a Frequency Modulated Continuous Wave radar (FMCW) with simultaneous transmission and reception



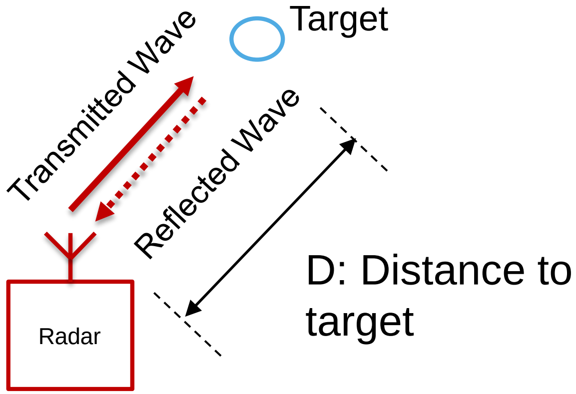
For the front LRR; increasing the aperture improves resolution; use of a Bistatic radar is one way to increase the aperture

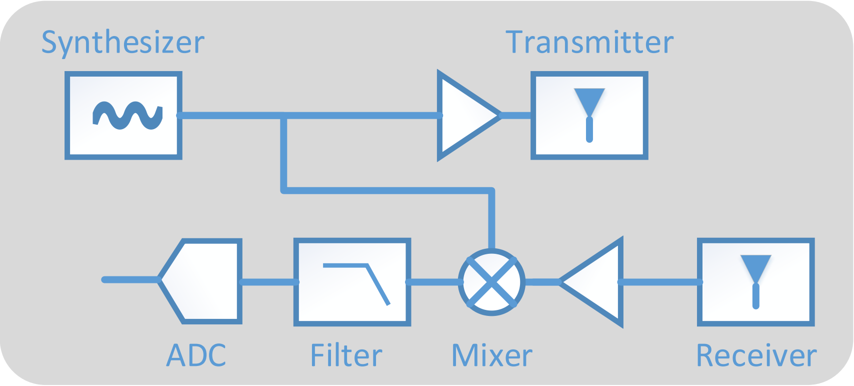
CHALLENGE: Synchronization between the radars

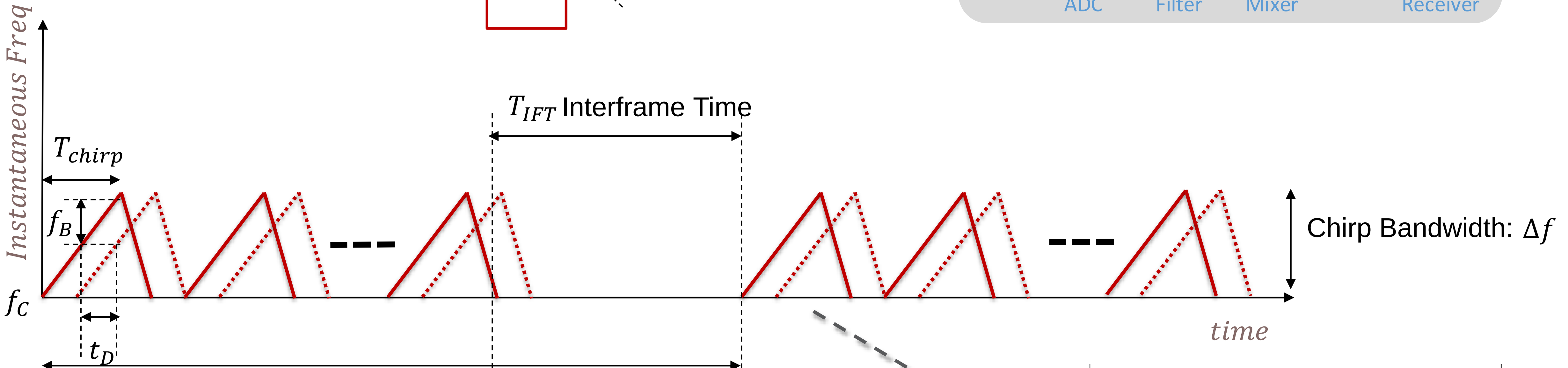
Figures not to scale

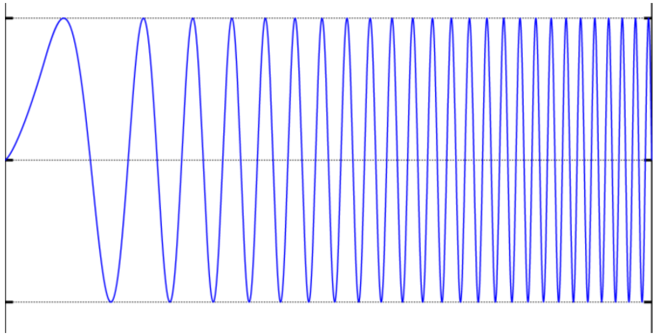
RADAR PRINCIPLES-2

Time of flight: $t_D = {}^{2D}/_{c}$









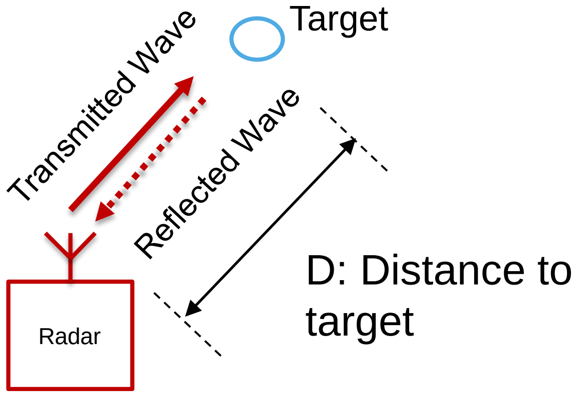
Chirped Signal (representational)

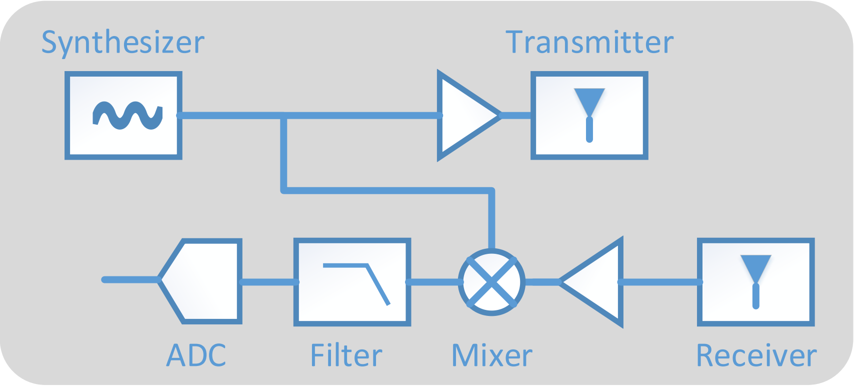
$$\frac{f_B}{t_D} = \frac{\Delta f}{T_{chirp}} \Rightarrow \qquad f_B = \left(\frac{\Delta f}{T_{chirp}}\right)\frac{2D}{c} \Rightarrow f_B \propto D$$

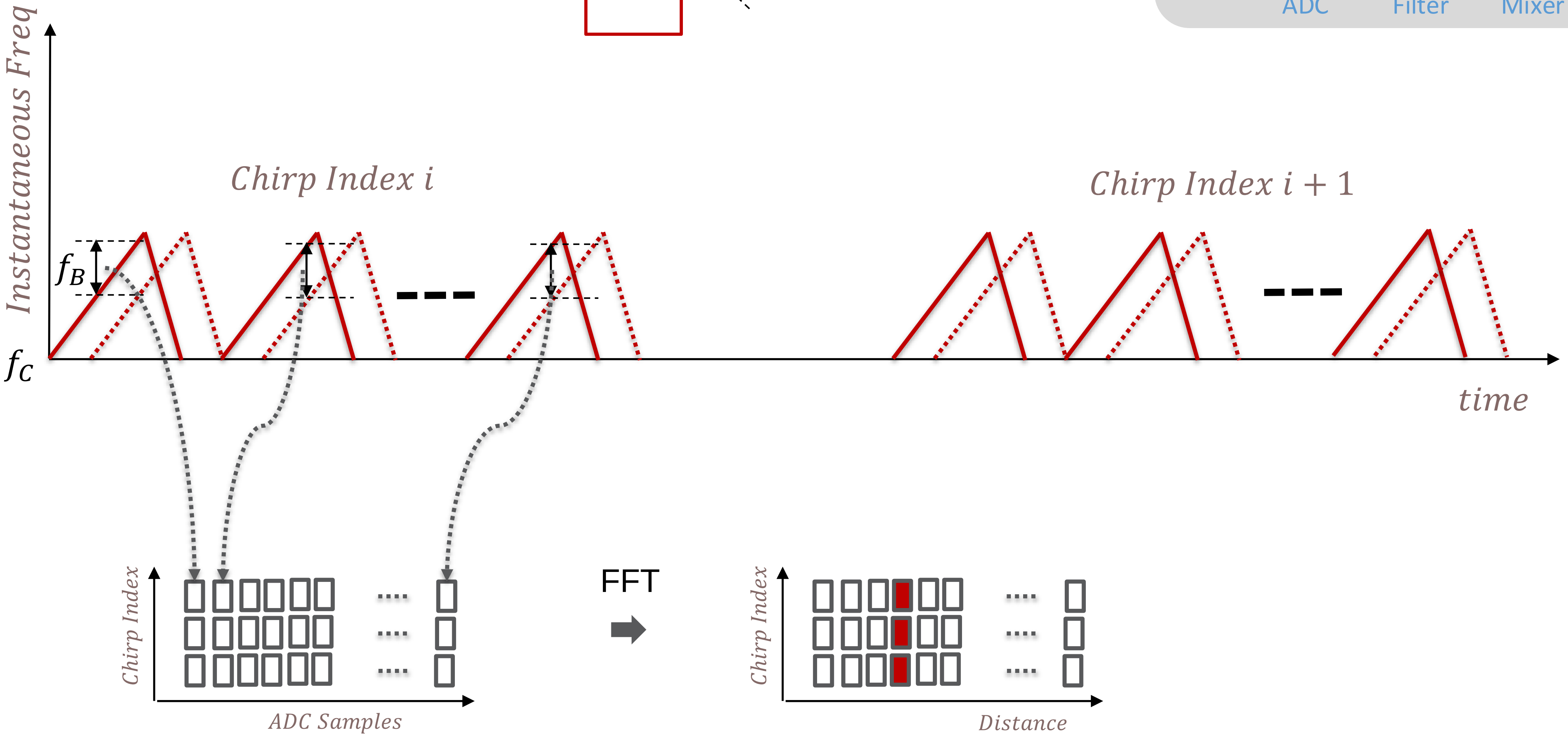
Beat Frequency is proportional to the target distance

RADAR PRINCIPLES-3

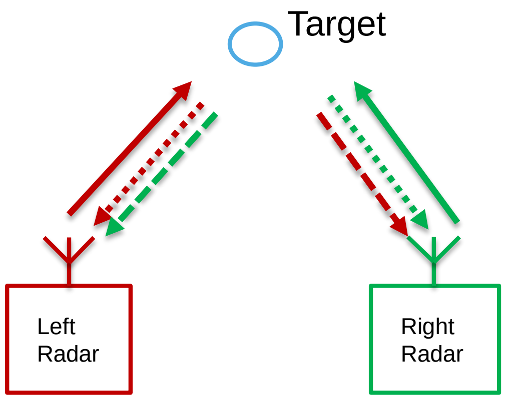
Time of flight: $t_D = {}^{2D}/_{c}$





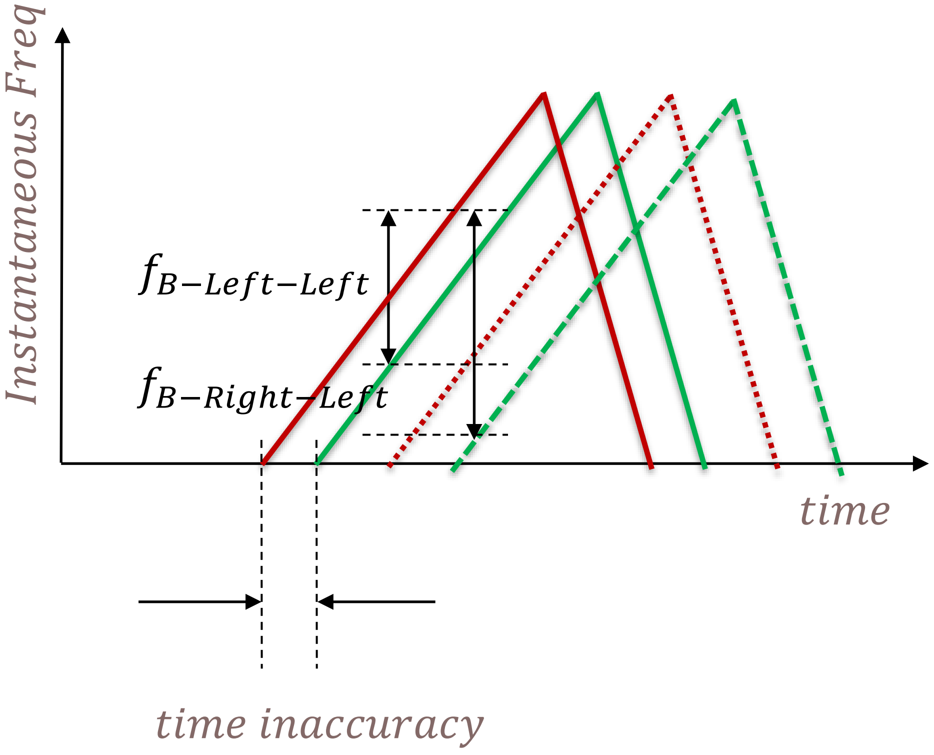


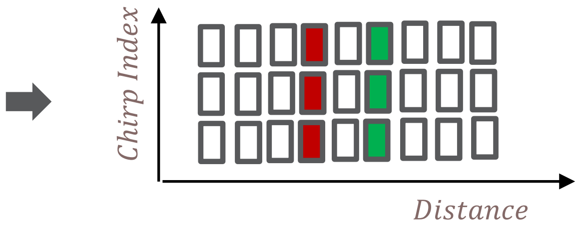
RADAR PRINCIPLES-4



$$f_B \propto D$$

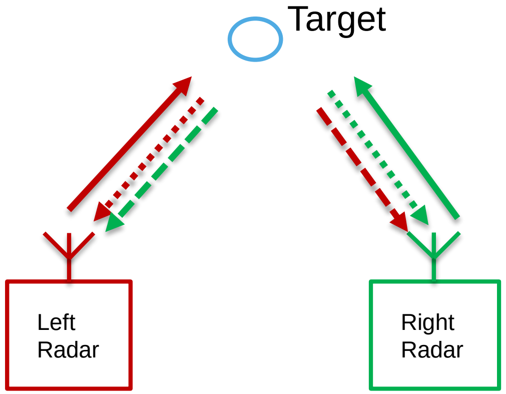
Beat frequency is proportional to target distance





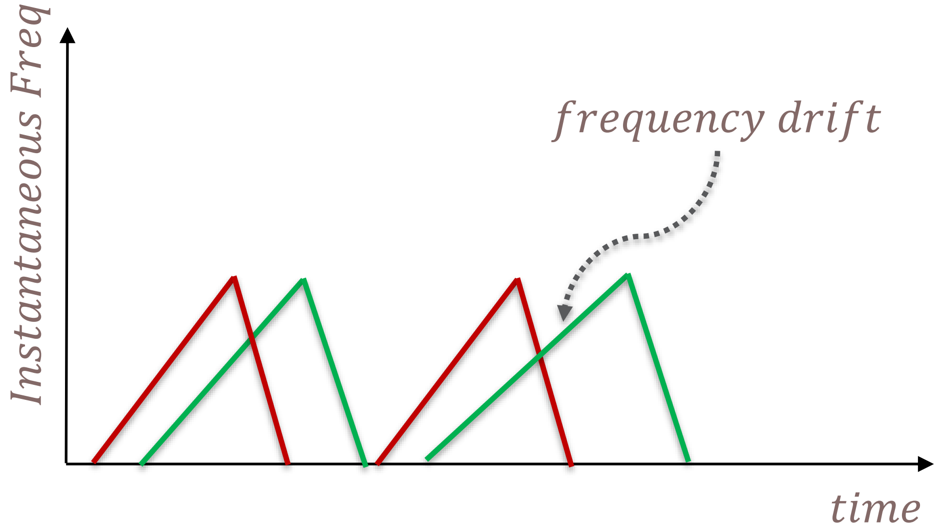
- Coherent Radars need to have the same notion of time of day to start the chirp sequence
- Time inaccuracy is directly related to the difference in estimation of distance to target
- The limit of the beat frequency is set by the downstream processing of the filter

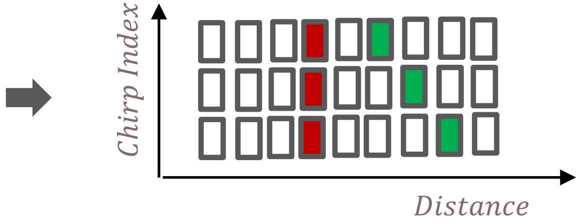
RADAR PRINCIPLES-5



$$f_B \propto D$$

Beat frequency is proportional to target distance





- Coherent Radars need have no relative frequency drift
- Frequency drift leads to the distance estimate changing and gives the appearance of a moving target
- The drift also limits the resolution of the target distances

Options to Improve gPTP's Accuracy

Status & Results of the 802.1's Industrial Profile's work

SECURE CONNECTIONS
FOR A SMARTER WORLD

Solving The *time inaccuracy* Problem

- TSN uses gPTP (IEEE 802.1AS) to get Precise Time-of-Day to Nodes in the Network
- The initial TSN use cases, Audio/Video, were addressed by AVB & IEEE 802.1AS-2011
- Does the AVB solution work for the Bistatic radar use case?
  - It can, depending upon the *time inaccuracy* a given implementation allows
  - Better accuracy is always better, especially if it’s added cost is close to free!

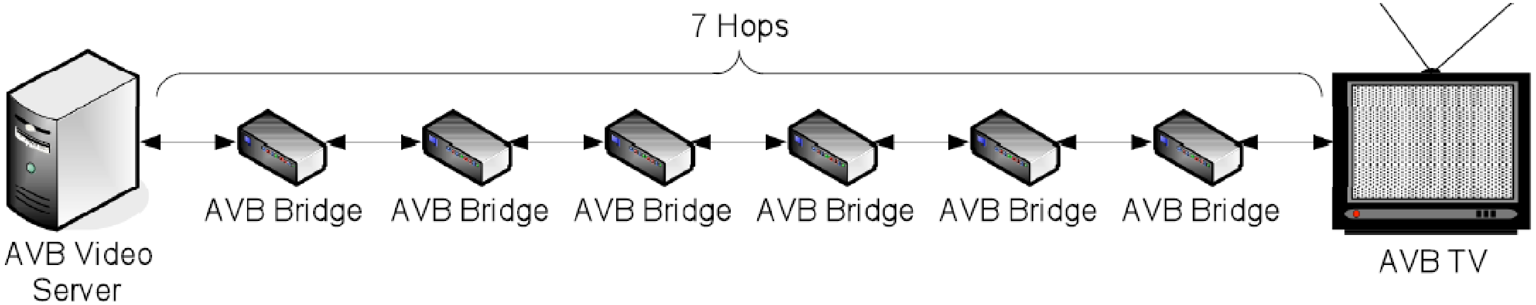
Good news:

- New Industrial TSN use cases require better accuracy than AVB & that work is nearly complete
  - A lot of work has gone into understanding the sources of gPTP inaccuracy much better

- Leveraging off this new TSN work gives a good indication of what can be achieved

Audio Video Bridging's Accuracy Goal (IEEE 802.1AS)

- AVB's accuracy for gPTP is +/-500ns between any two nodes over 7 hops (i.e., 1 Talker plus 6 Bridges) using 100 Mb/s links
  - This was achieved with:
    - Timestamp Granularity = 40 ns or 25 MHz
    - Pdelay Interval = 1000 ms or 1 Hz
    - Pdelay Turnaround Time = 10 ms
    - Sync Interval = 125 ms or 8 Hz
    - Sync Residency Time = 10 ms
    - Local oscillator quality = +/-100 PPM with <= 1 PPM/s drift
    - With a single hop accuracy of +/-71ns where the Timestamp Granularity is the largest component
      - See: https://www.ieee802.org/1/files/public/docs2009/as-garner-timestamp-accuracy-0109.pdf
      - And: https://www.ieee802.org/1/files/public/docs2010/as-garner-simulation-results-mult-replic-0910.pdf



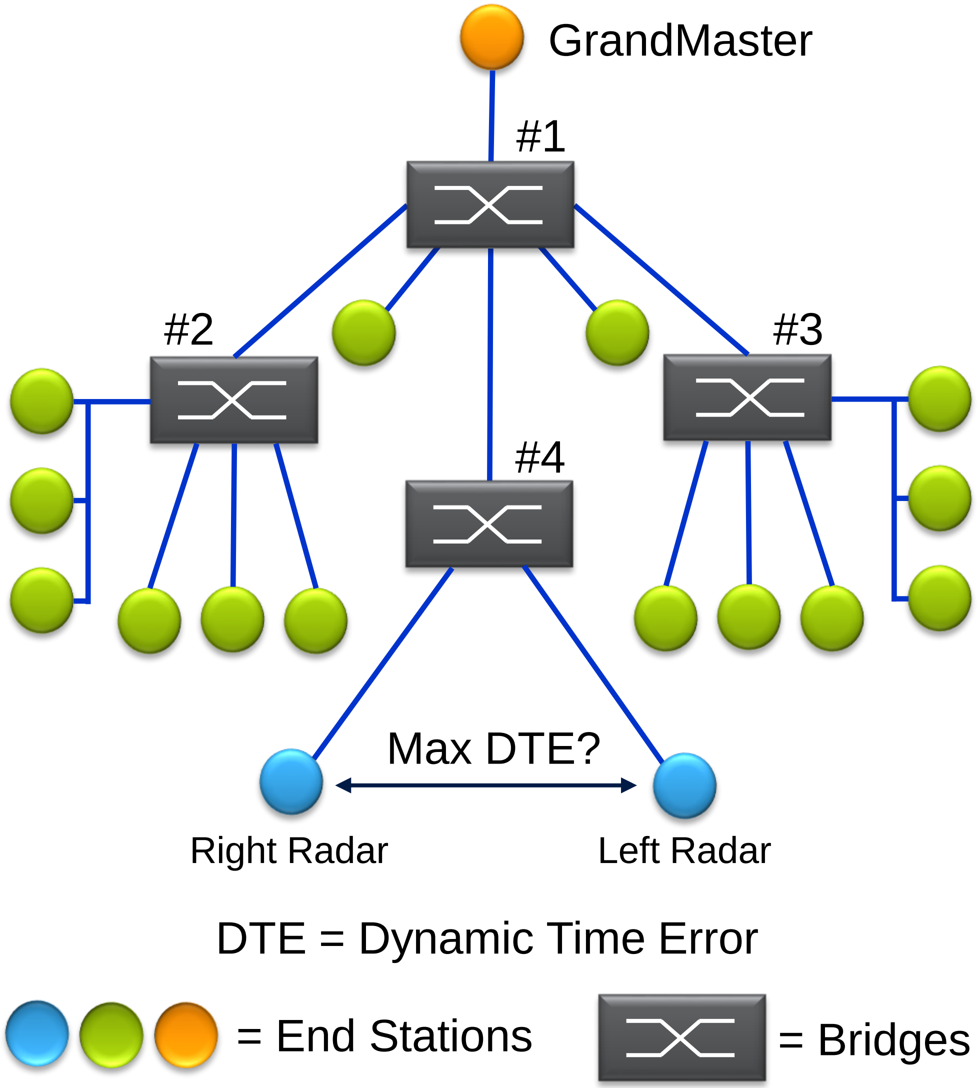
Industrial Automation’s Accuracy Goal (IEC/IEEE P60802)

- P60802 needs slightly less accuracy for gPTP, which is +/-1000ns between any two nodes, but it needs it with 100 hops! (i.e., 1 Talker plus a chain of 99 Bridges)
  - Is this even doable? A lot of work has gone into finding the most cost-effective way to achieve this goal without needing new silicon (i.e., what can be improved in the software?)
  - This appears to be achievable with (this is not quite finalized yet as the analysis is ongoing):
  - Changes from AVB are marked in Blue
    - Timestamp Granularity = 8 ns or 125 MHz vs. AVB’s 40 ns or 25 MHz
    - Pdelay Interval = 125 ms or 8 Hz vs. AVB’s 1000 ms or 1 Hz
    - Pdelay Turnaround Time = 10 ms
    - Sync Interval = 125 ms or 8 Hz
    - Sync Residency Time = 10 ms
    - Local oscillator quality = +/-100 PPM with <= 1 PPM/s drift
      - See: https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Time-Sync-Recommended-Parameters-Correction-Factors-0322-v04.pdf
- And there are other considerations / new discoveries too

Other Considerations / New Discoveries

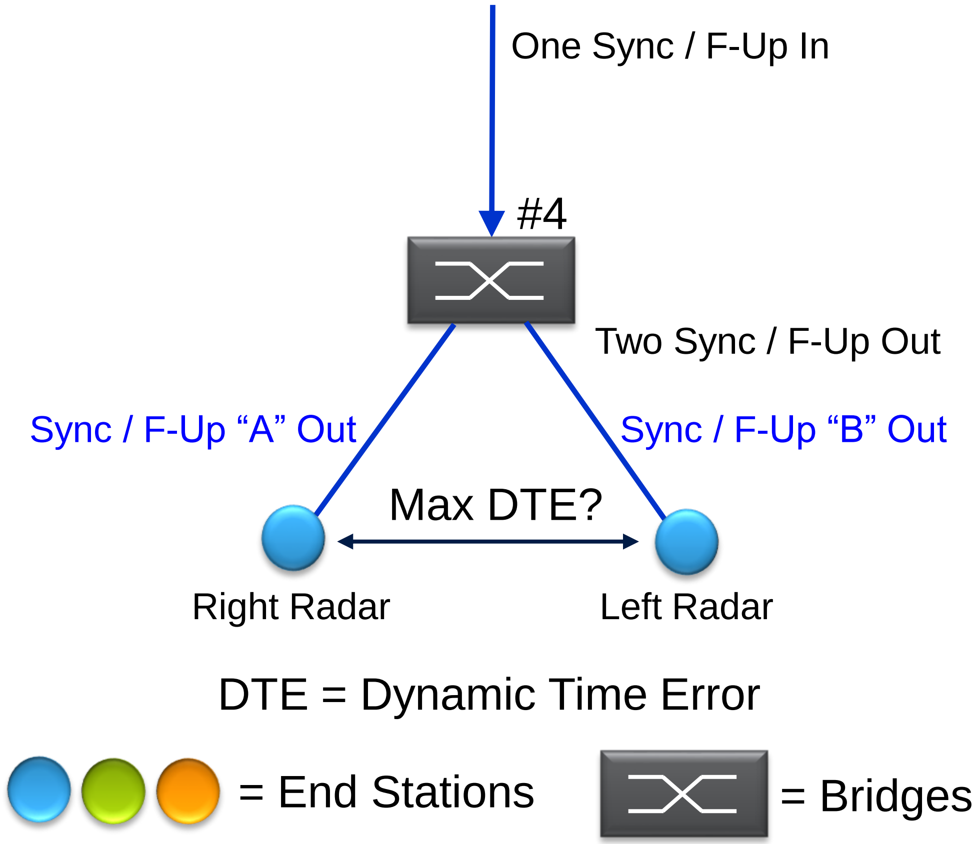
- Other considerations / new discoveries are:
- Averaging Pdelay measurements results in a meanLinkDelay with much lower error
  - A very low bandwidth, low-pass filter, is appropriate due the stability of Pdelay
  - This filtering gets Pdelay’s error to near zero!
    - See: https://www.ieee802.org/1/files/public/docs2021/60802-McCall-et-al-Time-Sync-Error-Model-0921-v03.pdf
- Residency Time can’t be averaged as its operational values are quite variable
- Neighbor Rate Ratio error occurs when clocks are drifting (e.g., due to temperature)
  - And can be minimized if the upstream sends Sync right after the downstream node completes its Pdelay exchange
    - Thus, the downstream node’s use of the Neighbor Rate Ratio is as close to where it is used (in the Sync’s residency time calculation) minimizing its error
    - A standard way to do this is to issue the Pdelay requests to the upstream node more often
    - Or the downstream node can anticipate when the next Sync is coming to issue its Pdelay request
- But are all these errors applicable to the Bistatic radar use case?

Focusing in at the Bistatic radar use case



- In this use case, the Dynamic Time Error of concern is between the two Blue End Stations only!
  - As these are the links that may need better accuracy between them, than what AVB gPTP can achieve
- Consider:
  - Bridge #4 gets a Sync & a Follow_up
  - Sometime later it forwards the Sync to the Right Radar
  - Sometime after that, is sends a Sync to the Left Radar
- The contents of the sent Sync frames are identical
  - And have the same error relative to the GM @ Bridge #4
    - Thus, any accumulated errors up to this point are irrelevant
- Remaining downstream error sources are:
  - Residency time difference between the two Syncs
  - Time stamp errors on the link (both ends)
  - Neighbor Rate Ratio errors in each Blue End Station

Looking at just the Bistatic radar’s connections



One Sync / F-Up In

#4

Two Sync / F-Up Out

Sync / F-Up “A” Out

Sync / F-Up “B” Out

Max DTE?

Right Radar

Left Radar

DTE = Dynamic Time Error

= End Stations

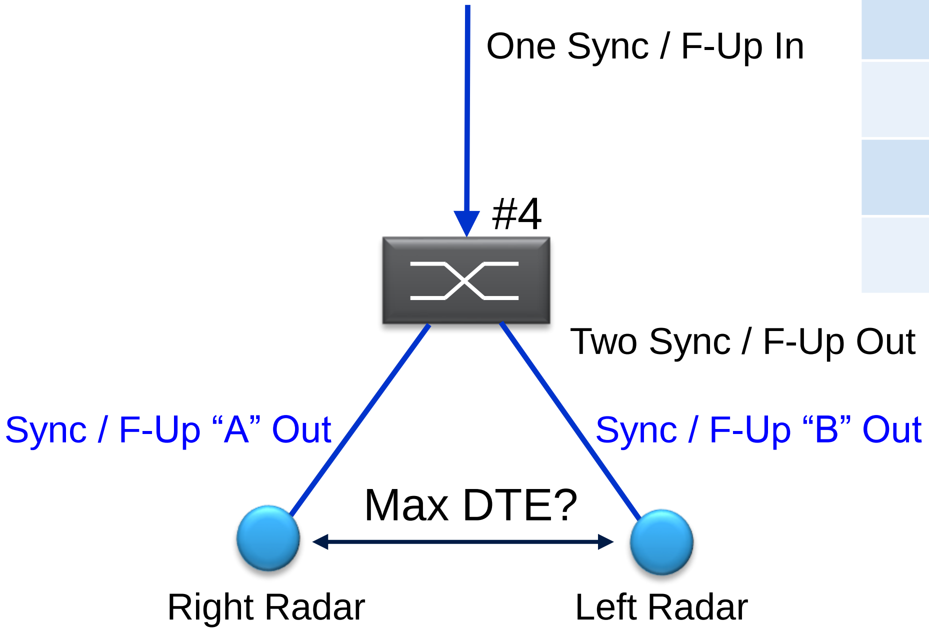
= Bridges

- David McCall worked on the math, and created an initial spreadsheet, to see what gPTP factors have the most effect on DTE for this focused use case
- This work is preliminary and needs more review, but the findings are logical and promising
- Since these are End Stations, worst case is assumed
  - Note: A chain of bridges have + & - errors which statistically cancels some of the error along the sync path
- Observations:
  - The largest source of error is clock drift, the change in ppm of a crystal over a 1 second interval
  - This creates errors in the Neighbor Rate Ratio as the crystal’s current ppm is not what it was when measured!
  - See: https://www.ieee802.org/1/files/public/docs2022/60802-McCall-Time-Sync-Errors-Complexity-Tradeoffs-Ad-Hoc-Next-Steps-0922-v02.pdf

Looking at just the Bistatic radar's connections

- Reducing ppm drift of a crystal can be expensive so what other options are there?

pDelayInterval	syncInterval	clockDriftLocal	Max DTE
1000 ms	125 ms	1.0 ppm/s	Baseline
125 ms	125 ms	1.0 ppm/s	5x lower error
125 ms	62.5 ms	1.0 ppm/s	10x lower error
125 ms	62.5 ms	0.5 ppm/s	15x lower error



DTE = Dynamic Time Error

● ● ● = End Stations ⌧ = Bridges

- The Baseline is the standard AVB gPTP setting
- All numbers assumes T1 PHYs with 8ns timeStamps
- Residency time difference between the two Syncs uses a transmission time difference of < 100 uSec
- Rounded relative Max DTE error comparison is used until more verification is done

Solving The *frequency drift* Problem

- AVB, together with IEEE 1722, solved the problem of multiple Listeners rendering identical media clocks
  - Needed to play the same song, in sync, on multiple speakers anywhere in the Network
  - Or for multiple (Talker) cameras at a sporting event capturing frames in sync so switching between cameras can be done seamlessly (i.e., on frame boundaries)
- IEEE 1722 supports identical media clocks across multiple End Stations using a very low bandwidth, Clock Reference Format
  - IEEE 1722-2016 Clause 10.1 Overview: "*The Clock Reference Format (CRF) allows for dissemination of event timing information to multiple TSN devices. CRF supports formats for audio media clock, video line, video frame, machine cycles, and other user-defined event timing*."
- CRF requires gPTP, and the more accurate gPTP is, the better
  - A solution for accurate gPTP between the two Bistatic radars was discussed above
- A tunable local PLL clock generator controlled by CRF completes the solution

Summary & Conclusions

Summary

- The timing needs for the Bistatic radar use case over TSN Ethernet has been discussed
  - *time inaccuracy* and *frequency drift* were identified as key performance areas to focus on
- The new work in IEEE 802.1 addressing lower *time inaccuracy* was discussed
- This new work can be applied to the Bistatic radar use case
  - Showing a worst-case DTE between the two Radars can be reduced significantly to:
    - 5x lower by only changing the downstream device’s Pdelay Interval to 125 mSec
    - 10x lower by also changing the upstream device’s Sync Interval to 62.5 mSec
    - 15x lower by also changing the downstream device’s crystal clock drift to 0.5 ppm/s
- It was suggested that IEEE 1722’s Clock Reference Format can support the needs of the *frequency drift* issue

SECURE CONNECTIONS
FOR A SMARTER WORLD