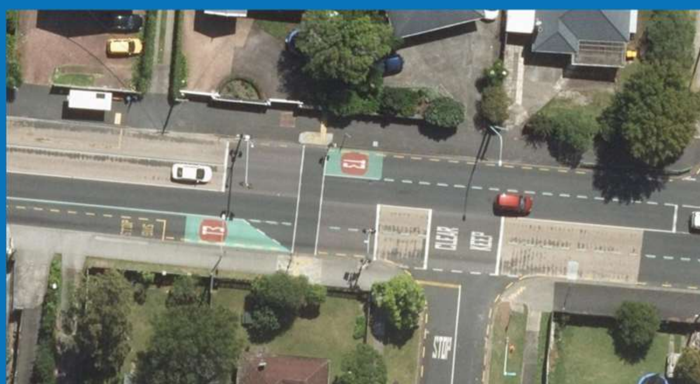
The humble mid-block pedestrian crossing



Rogan Parker
SNUG
September 2025

Think before you print

Thanks for giving me the opportunity to present here and I promise this presentation is worth listening to. This will take 15 to 20mins depending on how fast I babble, and there'll be time for questions at the end.
Feel free to jump in and stop me if there's something you want more clarity on. For those who want to try this in their own district, I'm happy to email the variation routines and the software files to get you going. My email address is on the final slide.

Starting off, how often do we look at our mid-block crossings. We commission them, leave them alone, occasionally optimise them as part of a corridor but generally don't pay them a lot of attention. They make up a good number of our signalised sites - 20% in Auckland - but do we really know how they perform?

We've had some great results from prioritising pedestrians at our midblock crossings and I believe it's applicable across New Zealand.

Picture is Onewa Rd by Wernham Place and it came to light in a Customer complaint in 2024.

This complaint resulted in a change to the way we operate our midblock crossings across

Auckland, and we've seen good reductions in pedestrian wait times.

Based on the results we've seen in Auckland, it's time to give your mid-block pedestrian crossings some attention.

Credit also to one of our ATOC engineers Jwee who has worked with me on this matter and has created new software to install our solution at new sites.

Firstly we'll look at how to measure pedestrian wait times, and then we'll look at how ATOC prioritises pedestrians.

Spoiler alert: this is a no (or very low) cost solution with 50+% reductions in pedestrian wait times at some sites.

Measuring pedestrian wait time

$$\text{Average Pedestrian wait time} = \frac{\text{Cycle Time} - \text{Walk time}}{2}$$

If Cycle Time is 60s, and walk is 6 secs:

Average pedestrian wait time = (60 – 6) / 2 = 27secs

SCATS records pedestrian demands and this can be accessed from event logs in SCATS History, but extracting the data and putting it into a useable format is time consuming and difficult. If the crossing is connected to SCATS, then we can generate pedestrian data and use SCATS History Viewer to interpret it.

We can use 2 methods to generate pedestrian wait time data:

- In software, MSS flag are triggered when there are pedestrian demands.
- Using variation routines to activate XSF flags when there are pedestrian demands.

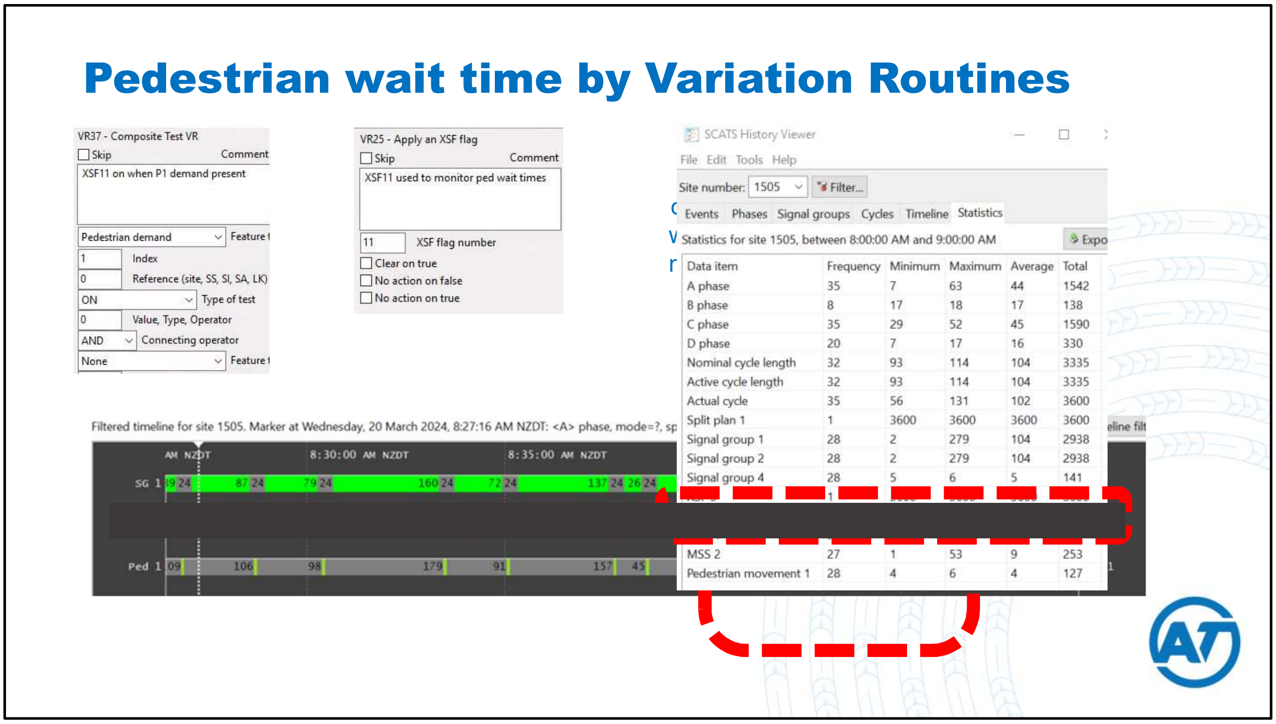
Note: This can be used at all intersections and isn't limited to mid-block crossings.

The first thing to consider is how do we measure pedestrian wait times.

We've probably all seen this old chestnut formula, which is really just a scientific wild arse guess

Pedestrian data is recorded in SCATS for every intersection with pedestrian crossings but it's difficult to extract and analyse.
We have spoken to Transport for New Soth Wales about getting better access to pedestrian data but as yet nothing has been developed by them, so we've created a simple workaround to generate the data at any existing SCATS monitored intersection and I mean intersections, not just midblock pedestrian crossings.

For new intersections and midblock pedestrian crossings, we build logic into the software to measure the pedestrian wait times.



On existing sites with SCATS monitoring, adding these 2 variation routines will generate the data needed.

The 1st variation is a simple test for a pedestrian demand and the second variation triggers the XSF flag.

Again this isn't limited to just mid-block crossings, we can use this at any intersection to collect pedestrian data.
Where there are multiple ped crossings, you will need the 2 variation routines for each pedestrian crossing.
For consistency, we are using XSF11-14 for P1 to P4 at intersections. In software we use flags MSS11-14 because software can only generate MSS flags and not XSF flags.

Once these variation routines are in place, the XSF flags automatically show in SCATS History Viewer allowing us to visualise the data

CLICK

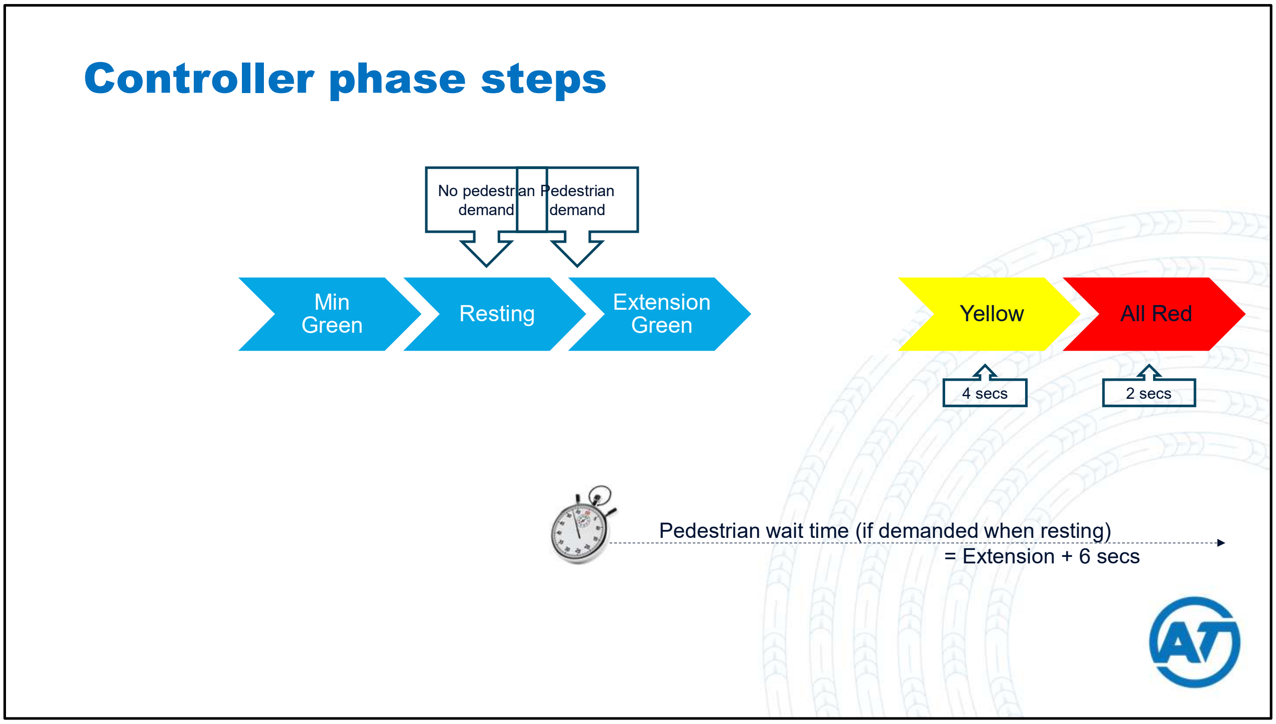
This is the Onewa Rd ped crossing before we made changes to its operation without the XSF flag showing. This is the Scats History Viewer "Timeline" tab. It shows us the vehicle green time Sg1 and the pedestrian green time P1 but doesn't give us any detail on pedestrian wait time. This is what you would see without the XSF1 data from the variation routines. Good information but not that useful.

CLICK
Adding in the XSF1 flag show us how long the pedestrian had been waiting for the pedestrian crossing .The XSF flag is on (green in the picture) once the button is pressed and goes off when the pedestrian crossing starts. This gives us a good visual of the pedestrian wait times compared to the vehicle green times.

CLICK
Highlighting a chunk, a pedestrian waited 30secs for their crossing when vehicles had a green light for more than 3 mins.
Another pedestrian waited 61secs when the vehicle green light had been on for 2.5mins.

Prior to having the XSF flag showing us the pedestrian wait time, the overall operation looked OK, now it's obvious the crossing needs some love.

WAIT then CLICK
XSF flag allows us to easily obtain the average pedestrian wait time and also the longest pedestrian wait time. This data is in the Scats History Viewer "statistics tab" and can be filtered if needed by time of day.
The risk as I see it, is the longer the wait time and especially the longest wait time, the more likelihood of jaywalking.



Time for a short bit of theory – promise it won't be enough time to have your after lunch nap.

To understand why wait times occur at midblock crossings, we need to look at how a controller steps through a phase. There are 7 different steps in a phase and the controller goes through each one sequentially.

We can ignore Late Start and Early cut off as these aren't used at midblock crossings. That leaves us with 5 steps.

CLICK

When there is no pedestrian demand

CLICK

the controller will rest until a pedestrian button is presses.

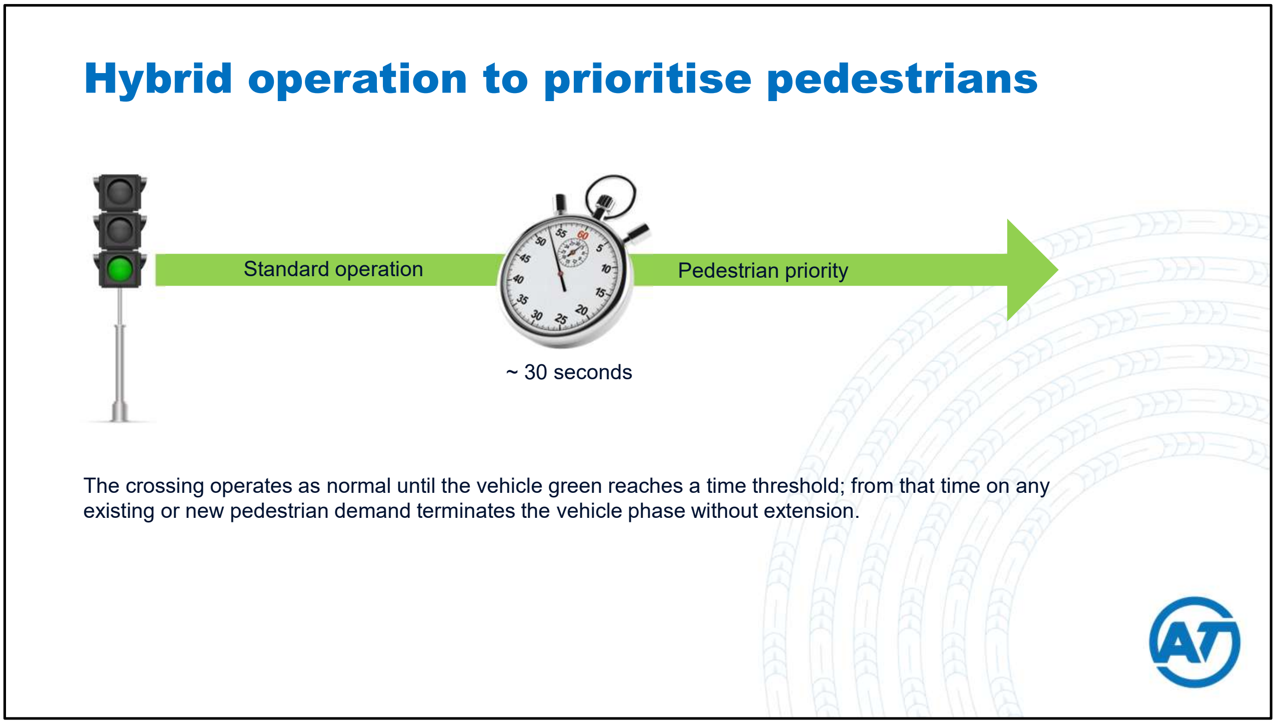
CLICK

Once the button is pressed, the controller steps into Extension Green and then the

Intergreen before entering the pedestrian phase.

CLICK

To summarise this, if the pedestrian demand is placed while the controller is resting, then pedestrian wait time is the sum of Extension and Intergreen (6secs).
An inherent issue with the way the controller works is that it can only look at what is happening at the time when the button is pressed and then after the button press. It can’t look back and see what’s happened previously.



We tried a number of ways to improve pedestrian wait times, lowering CT, double cycling, changing gap and headway timings, but those didn't work that well. Instead we use a hybrid operation.

After the previous pedestrian crossing, the vehicle signal turn green and the pedestrian crossing acts normally to start of with,

CLICK

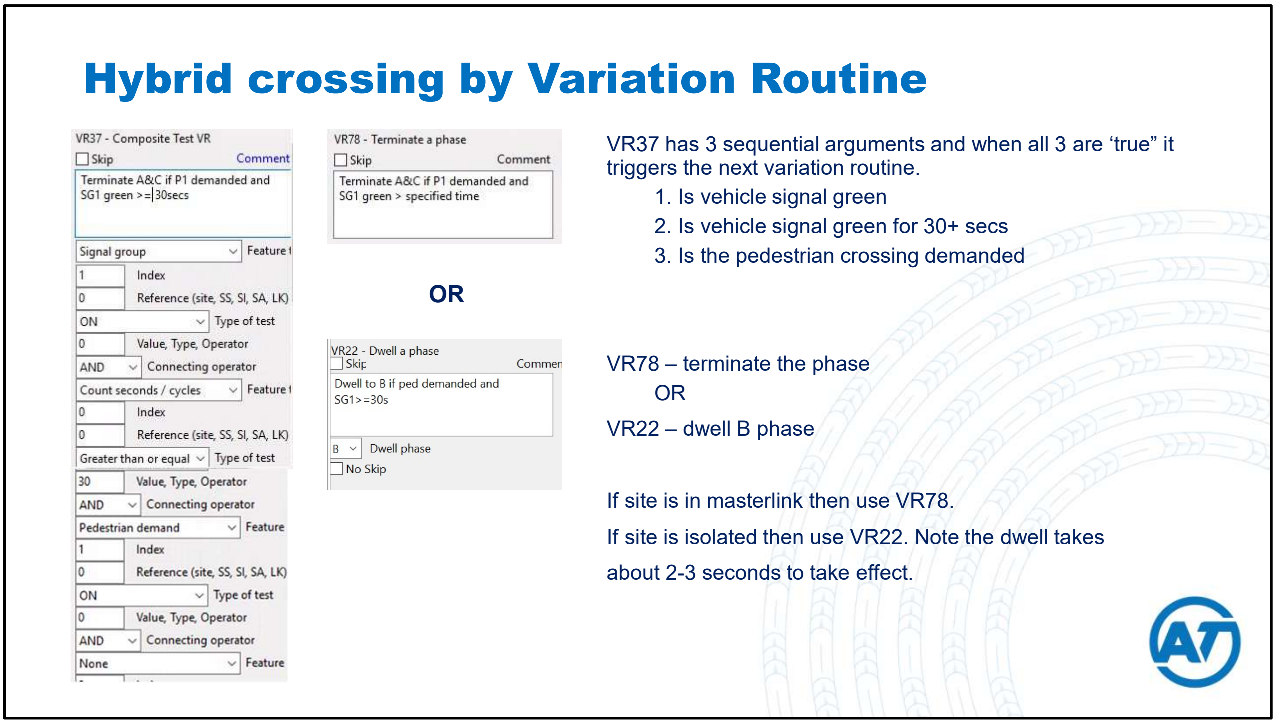
but after a set time and we usually use 30 secs

CLICK

We then alter the operation to prioritise the pedestrian crossing

Note that if the pedestrian demand is placed early in the phase (ie before the 30sec threshold, then the controller can move to the pedestrian phase when gaps are found in the traffic, and the hybrid operation does not take place.

So essentially, we have found a way to make the controller look back in time and use the previous 30 seconds to decide on when the pedestrian crossing will introduce



Remembering that ped wait time = extension +6secs, the way we prioritise pedestrian is to shorten the extension to its minimum.

There are 2 ways of setting up the hybrid operation: firstly using variation routines on existing sites monitored by SCATS. And secondly in software for new sites or sites being upgraded.

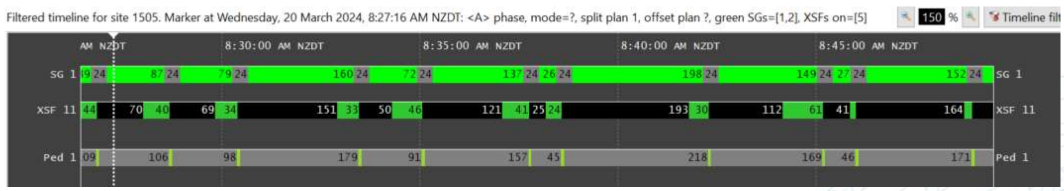
I'm focussing on the variation routines as this is a way engineers can try it on their own sites and see what difference it makes before going any further. I recommend putting in the wait time monitoring variations a couple of days (or a week) before the hybrid operation to get baseline pedestrian data.

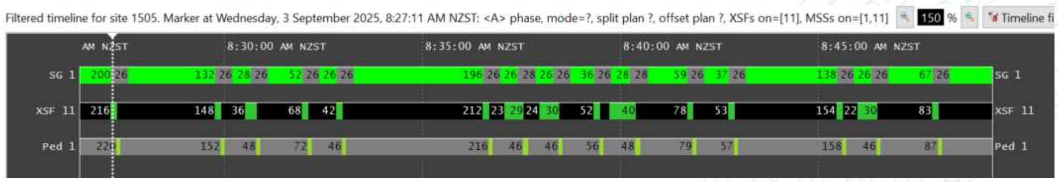
The first variation is 3 sequential tests. The first 2 tests check if the vehicle signal has been green for 30+ secs (or whatever time threshold you decide on) and the third test checks for the pedestrian demand
If all three are true then the second variation either terminates the vehicle phase or dwells to the pedestrian phase. If in masterlink/master-isolated then use terminate the phase. If isolated then VR78 terminate the phase won't do anything and you need to dwell to the pedestrian phase instead. Once controller goes into intergreen, then 1st variation becomes false and the dwell is removed.
Terminating a phase is fairly instant, but it takes a couple of seconds for the dwell to go

in.

In terms of the software Jwee has developed, we use A phase Max time as the threshold. The pedestrian priority logic is ON by default, but can be switched off remotely.

Results





Average pedestrian wait time (8.25am-8.50am) - down 53%

Longest pedestrian wait time (8.25am-8.50am) – down 33%

Lets look at some of the results.

At ATOC we've been using this system for the past 18months and a lot of our mid-block now run this pedestrian priority. As we come across mid-block pedestrian crossing through optimisation processes, CRM or faults, then we add the logic in. I'm guessing about 60% of our pedestrian crossings use this logic.

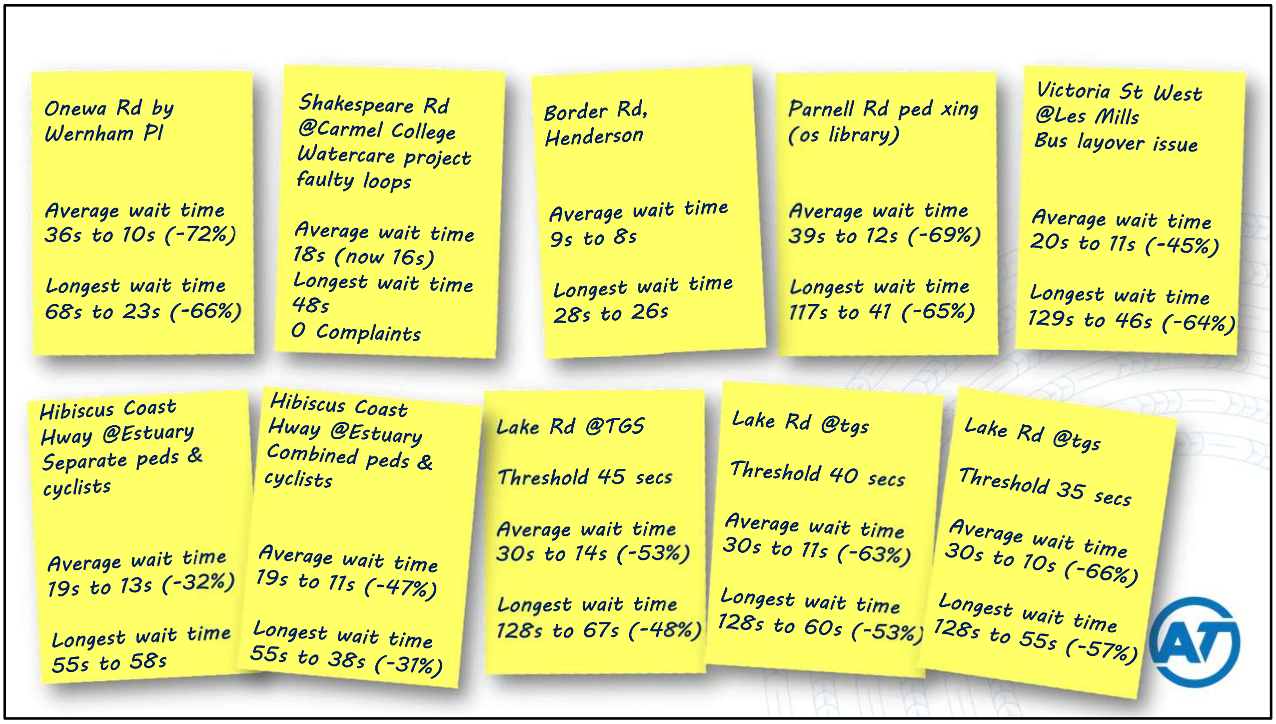
This is the Onewa Rd SCATS History viewer that we looked at before

CLICK

And the second picture is the same pedestrian crossing with our hybrid operation in effect.
Same day of the week, same time of the day, morning peak and just before school.

CLICK

In this 8.25-8.50am snippet, the average wait time dropped 53% and the longest wait time dropped 33%.



Here we have the results from a few sites around Auckland. The data is from a 24hr period on the same day of the week.

Onewa Rd (our first site) reduced average wait time by 72% to 10s and the longest wait by 66% to 23s. The is reflective of the congestion on the road and the peak time pedestrian demands. It's probably a best case scenario for the hybrid operation.

CLICK
Shakespeare Rd outside Carmel College was one we didn't have baseline data for. We installed the pedestrian priority when Watercare were doing a 6 month project on the road outside the school and we lost the in road detectors for that entire time. During this time, we had an average wait time of 18secs and longest wait time 48s, with zero complaints.
Now that the Watercare project is finished and detectors replaced, the average wait has dropped to 16secs. This shows how useful the hybrid operation is when we have faulty detectors.

CLICK
Border Rd is a worst case scenario for the hybrid operation. It is a medium to low use road with low pedestrian demand. There has been no real change to wait times as they are probably within the margin of error for sample variance.

CLICK
Parnell Rd is another best case scenario, A congested road with high pedestrian demand. We see 65 and 69% percent reductions in wait times.

CLICK
Victoria St West, Auckland CBD outside Les Mills saw a reasonable reduction in average wait time and a greater reduction to the longest wait time. The issue here is that buses layover near the crossing and occasionally park on the detectors so extend phases to maximum..

Click
Hibiscus Coast Highway crossing is an interesting example as this operated as a separate pedestrian and cyclist crossing. We saw a reduction in the average wait time but no real reduction in the longest wait time. This could be pedestrians pressing the cyclist button in error or pressing it while the cyclist crossing is running. We recently combined the crossings so that they run together regardless of demand and this has resulted in a reasonable reduction to the longest wait time.

Click
Lake Rd Takapuna Grammer School is another best case scenario. Lake Rd is heavily congested throughout the day and on weekends and the crossing is well used. Due to the congestion and the sensitivity of the road we started with a higher priority threshold at 45secs and have reduced that down over time to 35secs.
We see the longest wait time initially dropping substantially and then dropping by roughly the same time as the threshold reductions which makes sense .
The average wait time reduced initially be 53% and then by a further 13% when reducing the threshold.
Interestingly we don’t see a significant drop in average wait time between the 40 and 35s threshold.
Overall we are seeing reductions of 53 to 66% in average wait time de[ending on the threshold used, and likewise longest wait time reductions of 48 to 57%.

Conclusions

At pedestrian crossings on lower volume roads, a standard mid-block pedestrian can be expected to perform well. As vehicle volumes increase, average wait time increase.

Hybrid operation benefits:

- Improves wait times on roads with moderate to high traffic volumes.
- Improves wait times in peak traffic.
- Performs well at school crossings.
- Mitigates against faulty detectors.
- Easy to set up and can be added to existing sites.
- Low / zero cost

Hybrid operation issues:

- When a pedestrian button is “locked on”, the hybrid operation can be detrimental in higher volume traffic.
- Loss of Coordination in corridors

In conclusion, there’s some findings from our use of the hybrid operation pedestrian priority.

It has limited effect at sites with low traffic volumes or low pedestrian use. The heavier the traffic and pedestrian use, the more benefit it has.

At school crossings it finds a balance between pedestrians and clearing school vehicle traffic / school buses.

A detector fault means that the crossing will extend to max before running the crossing phase and can really damage pedestrian wait times. This operation caps those wait times.

Easy set up – 4 VRs in total , 5mins work per site.

No cost on masterlink sites. May be a cost on Isolated sites, but I haven’t been able to get a handle on what the cost actually is.

So the downsides:

Pedestrian alarm detection is an issue in itself. We (should I say Jwee) is currently working on new ways of detecting jammed buttons and then taking action when locked on. What Jwee is trialling is if button is ON for more than 2 or 3mins then an alarm is triggered and also the controller automatically increases Min Green and removes any pedestrian priority. If the button un-jams, the MIN Green returns to normal and pedestrian priority is switched back on.

If a corridor is coordinated, this operation can break that coordination. In this instance, it's an engineering decision as to whether to improve pedestrian priority or whether to retain corridor coordination. I still suggest installing the logic to measure pedestrian wait times as this will allow you to make an informed decision when considering the pedestrian level of service.

That’s it for me. I’m more than happy to email the variation routines, and we will also share the CIS and midblock software files with the logic written into them.

Don’t all jump at one. Questions??