A passive optical network, or PON, is typically used in a fibre-to-the-home setting. Optical signal from the network provider is split, without amplification or selection, to all users in the same network. The ONT here which sits somewhere in your home will select the data related to you.

* Typical usage of PON is in FTTH. However, gigabit PONs are already quite sufficient and mature with low cost hardware.
* The motivation to go up to 100 Gb/s is to share other applications along existing fibres, to reduce the cost of installing new fibres.
A particularly interesting application is in mobile networks. As people get more addicted to their smart phones, mobile network operators would need to increase the density in cell sites, making PON a good candidate to deliver the cell site backhaul.
As 5G mobile develops, PON may also be useful in the fronthaul. RF receivers relay the signals back to a centralized location for processing, which is seen as a way to reduce operating costs.

To achieve a high data rate, coherent receivers are to be used. This is different from the direct detection receivers which you may have come across last year in 3B6, where the receiver photocurrent is proportional to the optical power. In a coherent receiver, as illustrated by this example here, both the real and imaginary parts of two orthogonal polarizations are detected. This allows different modulation schemes, similar to radio transmissions, such as PSK and QAM.

In this project, simulation models for optical networks will be built using MATLAB, with digital signal processing to correct for fibre effects. Different options for achieving the target data rate of 100 Gb/s will be simulated and compared, and later experimentally validated. The results will be evaluated in terms of suitability to use in a PON.

The overall model has this structure, with root-raised cosine pulses used. The channel is modelled as additive white Gaussian noise, followed by the simulated physical effect.
On the receiver side, after analog-to-digital conversion, the received complex symbols r_n are further processed to compensate for channel effects before decision and demodulation.

Currently I have based by simulations on a quadriphase-shift keying modulation scheme with 25×10^9 symbols per second. QPSK gives two bits per symbol, giving 50 Gb/s. Adding in polarization-division multiplexing, which I haven't done yet, would reach the target of 100 Gb/s.

* The first effect investigated was chromatic dispersion. This effect occurs as a result of the speed of light being slightly different at different wavelengths, and lasers have a wavelength band that, although small, still makes a large impact over long distances.
Literature has shown that chromatic dispersion can be modelled as a linear system, with D being the dispersion coefficient and z the distance travelled.
* In direct detection receivers, this effect can be seen as a pulse-broadening effect. However, in coherent receivers, we are more interested in the changes to the complex constellation symbols.
* Here is a result of a simulation with 17 ps/(nm km) dispersion, with 1 kilometre of fibre, in the absence of any additive noise. You can still cleanly decode the symbols without much difficulty.
But when we go slightly longer to 5 km...
* we get this mess.

So clearly the receiver needs to do something to mitigate the effects of chromatic dispersion. A linear filter can be used. How do we design this filter? Well, we know the impulse response of the dispersion model, so if we invert the sign of D here...

* We get the impulse response of the dispersion compensating filter.
* Additive noise was added back to the channel, and a million bits were sent through the channel. This is a plot of the bit-error rate, or the probability of decoding a bit incorrectly, against a measure of the signal-to-noise ratio, at a simulated transmission distance of 200 km. We can see that while the magenta curve, that is without any compensation, doesn't do any better than chance, The red curve, which is the simulation result of the compensating filter, does a very good job at approaching the theoretical blue curve of an ideal AWGN channel.

* An interesting behaviour was observed when the transmission distance was reduced, in this case, to 2km. We can see that the compensation filter actually does worse, which is counter-intuitive. This is due to a reduced number of filter taps when converting this continuous-time filter to a discrete-time filter. We can, of course, add an extra filter to try to correct for this, which brings us to the next topic:

Adaptive equalization. Here, the error of the previous symbol is used to update the filter taps, which can correct for static as well as slowly varying effects. In the following simulations, the constant modulus algorithm was implemented. This relies on the fact that, while the receiver doesn't know what symbol was transmitted, it knows, for PSK, that the symbols must lie on a unit circle. The distance from the received signal to the unit circle is thus used as a measure of error.

* Here is an animation of how the adaptive equalizer converges, again with a small dispersion but without additive noise. Initially the symbols are quite widely spread, but as the algorithm runs, /just click through the slides/
we can see the equalizer brings the symbols close to 4 single points.
* The overall effect can be seen with additive noise added back in, here with the green curve very closely agreeing with the theoretical blue curve.

The next effect investigated was laser phase noise. This arises as the true laser wavelength is a slight deviation from the nominal central wavelength within the linewidth. This instantaneous change in wavelength can be modelled as a phase shift. Literature suggests a one-dimensional Gaussian walk model, where each sample of the phase differs from the previous sample with a random delta phi drawn from a Gaussian distribution, with zero mean and whose variance is proportional to both the linewidth delta nu and the sampling time T_s.

The effect of phase noise is a rotation of the constellation symbols by an arbitrary amount, which is very problematic for PSK modulation as this would mean completely incorrect decoding.

We will consider two methods to mitigate this effect. The first is to use a differential PSK scheme, where the information is encoded not as the absolute phase, but as a difference in phase with the previous symbol. This works relying on the assumption that the phase noise between two consecutive symbols is sufficiently smaller than 90 degrees. However, by considering two symbols together, the effect of phase noise is enhanced, translating to a penalty in the signal-to-noise ratio.
Another method is to estimate the amount of phase noise in each symbol and to rotate the symbols back by the estimated amount. The Viterbi-Viterbi algorithm, very briefly, works by taking an average over a block of samples to estimate the average phase noise in that block.

This estimation algorithm does not always work though. Sometimes, due to noise, a cycle slip would occur, which results in a systematic error of 90 degrees. This means that all the following symbols would actually be decoded incorrectly.
* We can see in this simulation result that, when a cycle slip does not occur, the performance is much closer to the ideal curve than using differential PSK, but when it does occur the result is disastrous.
Can we get the best of both worlds, with a small penalty but without any cycle slips?
* The answer is yes, with a differential *encoding*. Note that this is different from differential PSK. Differential encoding is an operation on the bits, and the receiver would perform differential decoding on the bits *after* choosing the closest constellation symbols. Whereas in DPSK the receiver evaluates the difference in phase between the received samples directly, before converting them to bits.
* The result is this cyan curve, with a smaller penalty than DPSK, and not affected by cycle slips.

That's all I have done as of now. Looking forward, simulations would need to integrate both effects into a single channel. Further investigations into adaptive equalizers can also be done. For example, the use of decision-directed algorithms, and to train the equalizer with pre-known sequences, to increase accuracy at the cost of losing some data rate.
Other modulation schemes can also be investigated, such as higher-order QAM. It would require careful alterations to existing implementations, for example with the constant modulus algorithm, but can reduce the required symbol rate.
The simulations performed only considered a single polarization. Further investigations should be done to send data through one more polarization, and the resulting effect of polarization mode dispersion needs to be addressed. Adaptive equalizers for PDM signals also need to be revised.
Finally, non-linear fibre effects can also be considered.
All these should hopefully be done before mid-Lent, leaving sufficient time for experimental measurements.

That's all I've got. Thank you.