Università degli Studi di Padova
Dipartimento di Ingegneria
dell'Informazione
Corso di Laurea Magistrale in
Ingegneria Informatica

Sviluppo di protocolli di sincronia per la Quantum Key Distribution

Relatore:
Prof. Giuseppe Vallone
Co-Relatori:
PhD Giulio Foletto
PhD Davide Scalcon

Laureando:
Nicola Prevedello
matr. 1236454

Anno Accademico 21/22
Data di laurea 14 luglio 2022

Sommario

La crittografia è la scienza che al giorno d'oggi ricopre il ruolo fondamentale di garante della privacy della nostra identità digitale. Questa tecnologia negli ultimi tempi, per poter stare al passo con lo sviluppo di Internet, ha dovuto affrontare un incessante processo di innovazione che l'ha portata a mutare i meccanismi di base del proprio paradigma. L'evoluzione dei mezzi di comunicazione ha spinto la crittografia all'uso di numerosi approcci differenti, passando sia per metodi classici che per procedure più moderne. Recentemente però l'attenzione è stata rivolta verso le possibilità offerte dal modello quantistico.

In questa tesi viene presentata la mia esperienza all'interno dei laboratori dell'università di Padova, dove in collaborazione con il gruppo Quantum Future ho potuto partecipare allo sviluppo del progetto per la Quantum Key Distribution (QKD). Questa tecnica, che si fonda su importanti teoremi di fisica quantistica, è in grado di assicurare uno scambio sicuro di informazioni, azzerando la possibilità di intercettazione da parte di terzi.

In seguito alla parte introduttiva all'argomento, questa tesi approfondisce le problematiche legate alla sincronizzazione dei dispositivi coinvolti nella comunicazione quantistica. Vengono proposti vari protocolli che utilizzando risorse differenti (il segnale quantistico, un riferimento esterno, il segnale fornito dal GNSS) mirano tutti a realizzare la sincronia tra trasmettitore e ricevitore. Nella parte finale di questo elaborato vengono raccolte ed analizzate le performance ottenute dai vari protocolli.

Indice

Capitolo 1

Introduzione

La necessità di scambiare informazioni in sicurezza è sempre stata presente nella storia del uomo ma con lo svilupparsi della tecnologia raggiungere questo obiettivo è diventato sempre più difficile. Per questa ragione la crittografia, ossia la disciplina che studia come rendere un messaggio comprensibile solo alle persone direttamente coinvolte nella comunicazione, si è dovuta evolvere, passando da tecniche elementari ed arrivando a costruire stratagemmi molto più complessi sostenuti alla base da solide teorie matematiche.

Questa scienza, la cui applicazione in principio era limitata all'ambito commerciale e militare, ai giorni nostri ha subito una massiccia diffusione causata dall'avvento di internet, ed è così entrata a far parte della nostra vita quotidiana. Nella nostra società infatti la crittografia ricopre un ruolo fondamentale impedendo che informazioni personali e più in generale il flusso di dati che ogni giorno viene immesso sulla rete possa essere accessibile da chiunque. In questo modo riusciamo ad avere la certezza che le informazioni che condividiamo e che escono dal nostro controllo rimangono private, o quasi. Quella della crittografia infatti non è una teoria che può dirsi conclusa, assieme alla crittografia è nata la sua antagonista la crittoanalisi, la scienza che per l'appunto si occupa di svelare i messaggi nascosti dai codici della crittografia. Lo sviluppo di una spinge al progresso l'altra e in questo modo si procede senza sosta tentando di mantenere segrete le informazioni che dovrebbero restare private.

Uno dei primi metodi per nascondere un messaggio è stato quello di usare il cosiddetto *Cifrario di Cesare*[1]. Questa tecnica consiste nel sostituire ogni

[1]Il *Cifrario di Cesare* è uno dei primi e più importanti metodi crittografici, questa tecnica venne largamente usata al tempo del generale romano da cui prende il nome, il suo successo è principalmente dovuto al fatto che all'epoca non esistevano efficaci strumenti di crittoanalisi ed inoltre già soltanto la capacità di lettura di un normale testo non era molto comune.

lettera del messaggio con quella che si trova K posti più avanti nell'alfabeto. In questo modo esso diviene incomprensibile e può essere letto soltanto da chi conosce la chiave ossia il destinatario che sa quale valore di K è stato applicato al contenuto del messaggio. In seguito, l'invenzione da parte di Al-Kindi, un noto matematico arabo, dell'analisi delle frequenze rese il Cifrario di Cesare una tecnica obsoleta e spinse lo sviluppo verso tecniche alternative come il *Disco Cifrante* di Leon Battista Alberti. Questa tecnica che alla base conserva lo stesso meccanismo del Cifrario di Cesare si distingue per non rientrare più nella categoria dei metodi monoalfabetici ma per essere il primo rappresentante delle tecniche polialfabetiche[1]. In questo modo la stessa lettera contenuta nel messaggio poteva venire cifrata una volta in un modo e la successiva in uno diverso, riuscendo così a rendere l'applicazione dell'analisi delle frequenze più difficile.

Uno dei più importanti rappresentanti di questo genere di metodi crittografici appartenenti alla crittografia classica anche detta crittografia a chiave privata è Enigma[2]. Questa macchina elettromeccanica originariamente applicata in campo commerciale e successivamente presa come riferimento dall'esercito tedesco durante la seconda guerra mondiale rese le comunicazioni tedesche inviolabili. Soltanto grazie alla cooperazione dei più brillanti matematici inglesi e polacchi fu possibile realizzare quello che in seguito sarebbe stato riconosciuto come uno degli antenati del computer moderno, ossia Bombe una macchina in grado grazie ad un attacco a forza bruta di violare il codice di Enigma rendendo chiare e leggibili anche dagli Alleati le comunicazioni tedesche. Il progresso fatto da questi crittoanalisti mise in chiaro quali erano le debolezze degli approcci usati all'epoca ed evidenziò ancora una volta l'importanza dello studio della crittografia. In seguito venne stimato che la rottura del codice di Enigma da parte degli Alleati accorciò la durata della guerra di almeno un paio d'anni.

Gli aspetti critici che erano emersi dalla vicenda di Enigma avevano messo alla luce le debolezze di questi approcci classici basati su codici polialfabetici a scambio o scorrimento. Molta della loro complessità infatti era ottenuta tenendo segreta la struttura interna della macchina e aumentando a dismisura le possibili configurazioni iniziali che questa poteva assumere. Con l'arrivo del computer però ci si rese conto che non erano questi gli aspetti che dovevano essere responsabili per la sicurezza di un codice, infatti come fu stabilito da Kerckhoffs e riformulato da Shannon: *"un sistema crittografico deve essere sicuro anche se il nemico conosce i dettagli realizzativi del sistema stesso"*.

Seguendo questa linea di pensiero venne realizzata la *crittografia RSA*[3], conosciuta anche come crittografia a chiave pubblica. La particolarità di questo metodo sta nel non richiedere uno scambio di informazioni tra Alice e Bob per la costruzione della chiave: questo sistema fa infatti uso di due

chiavi, una detta "chiave pubblica" e una chiamata "chiave privata" e utilizza un metodo di cifratura asimmetrico. In un sistema a chiave asimmetrica la chiave usata per cifrare il messaggio e quella usate per decifrarlo non coincidono, in questo modo è possibile per Alice pubblicare la chiave usata per cifrare il messaggio (la sua chiave pubblica) e mantenere segreta quella usata per decifrarlo (la sua chiave privata), restando di conseguenza l'unica a poter decifrare i messaggi a lei diretti. Se un utente volesse iniziare una comunicazione usando la crittografia RSA dovrebbe cifrare il proprio messaggio usando la chiave pubblica del destinatario, in questo modo l'utente che riceve il messaggio sarà in grado di decifrarlo usando la propria chiave privata. La sicurezza che questo sistema è in grado di garantire è ottenuta sfruttando una funzione unidirezionale basata sull'aritmetica modulare. Alla base del sistema infatti ci sono importanti trasformazioni matematiche che facendo uso di questa funzione ed in particolare della relazione che questa ha con i numeri primi, permettono di generare le chiavi da assegnare agli utenti permettendo di rendere i loro messaggi incomprensibili e con la stessa facilità invertire il processo. L'unico modo che si ha per rompere il sistema sarebbe quello di riuscire ad identificare la relazione che lega la chiave privata a quella pubblica. Per fare ciò è necessario riuscire a trovare una fattorizzazione della chiave pubblica, un problema che è risolvibile soltanto in un tempo non polinomiale e quindi non affrontabile con le risorse in commercio attualmente.

Con il passare degli anni però nuove tecnologie vengono sviluppate e nuovi algoritmi vengono ideati, per mantenere sicuri i metodi crittografici attuali aumentare la dimensione delle chiavi utilizzate risulta essere una soluzione solamente temporanea. I grandi colossi digitali come Google e IBM hanno già iniziato a cimentarsi nello sviluppo di computer quantistici in grado di offrire possibilità che con le risorse attuali erano inaccessibili. Ad esempio, l'algoritmo di fattorizzazione di Shor[4], grazie all'architettura quantistica sarebbe in grado di abbassare il costo computazionale della fattorizzazione di numeri interi rendendolo polinomiale, facendo risultare vulnerabili la maggior parte delle tecniche crittografiche attuali. Per questo motivo si stanno cercando soluzioni alternative in grado di preservare la sicurezza delle tecniche crittografiche, e garantire che il progresso e le innovazione tecnico-scientifiche non sovvertano il sistema attuale. Soluzioni di crittografia quantistica stanno venendo studiate ed in base ad una serie di teorie, le quali verranno discusse più avanti, si sono rivelate essere interessanti e promettenti in quanto dovrebbero riuscire a garantire l'assoluta inviolabilità del sistema.

Negli ultimi anni hanno suscitato notevole interesse le tecniche di Quantum Key Distribution (QKD) che sfruttando nozioni di fisica quantistica sono in grado di permettere lo scambio della chiave tra Alice e Bob in assoluta sicurezza, eliminando la possibilità che questa possa essere intercettata da

utenti malevoli. Queste tecniche per poter trasmettere l'informazione contenuta nello stato quantistico fanno uso di singoli fotoni, che grazie alle loro proprietà possono facilmente essere trasmessi su fibra ottica. A causa dell'elevato livello di precisione richiesto per la rilevazione del segnale quantistico, assieme all'implementazione di queste tecniche viene portato avanti lo sviluppo di protocolli di sincronia in grado di minimizzare il valore delle perdite del segnale, alzando allo stesso tempo l'efficienza dell'intero protocollo.

Questa tesi si propone di illustrare quella che è stata la mia esperienza all'interno dei laboratori dell'Università di Padova diretti dal gruppo QuantumFuture, nei quali ho avuto la possibilità di partecipare al progetto di Quantum Key Distribution. Il mio ruolo all'interno del progetto è stato quello di studiare il protocollo di sincronia Qubit4Sync già precedentemente implementato dal gruppo e realizzare la progettazione e l'implementazione software di protocolli alternativi che sfruttando la disponibilità di risorse aggiuntive fossero in grado di produrre buone performance discostandosi dallo schema di funzionamento del protocollo già esistente.

Segue una breve rassegna del materiale che verrà affrontato nei seguenti capitoli:

- **Quantum Key Distribution:** Viene discussa la tecnica della QKD, protagonista di questa tesi, approfondendo quali sono state le sue origini e quali sono i teoremi fondamentali che sostengono questa tecnica.

- **Protocolli di sincronia:** In questo capitolo viene spiegato che cos'è un protocollo di sincronia e quali parametri è necessario ricavare per poterne realizzare uno. Vengono presentati i due principali protocolli con cui si ha avuto a che fare per la realizzazione di questa tesi ossia il Qubit4Sync e l'ExtRef. Infine vengono illustrate le tecniche principali di allineamento introducendo inoltre un metodo che permette di usare una stringa di sincronia casuale condivisa al momento dell'esecuzione, aprendo le porte a metodi di QKD basati sull'entanglement quantistico.

- **Analisi dei risultati:** In questo capitolo viene riportata un'analisi statistica delle performance dei protocolli precedentemente riportati. Andando a simulare un possibile scenario di applicazione l'analisi ha lo scopo di verificare la validità degli approcci alternativi al Qubit4Sync.

- **Conclusioni:** Vengono riassunti i risultati ottenuti con lo sviluppo degli approcci alternativi riportati in questa tesi dando anche uno sguardo a quello che potrebbe essere il futuro della QKD.

- **Gestione del progetto:** In questo capitolo vengono discusse alcune scelte di design del progetto, riportando le ragioni per cui si è scelto

questo tipo di realizzazione. A questo scopo verranno riportati anche gli applicativi usati per lo sviluppo motivandone l'utilizzo.

Capitolo 2

Quantum Key Distribution

Per comprendere al meglio che cosa ha spinto l'evolversi delle tecniche crittografiche verso lo sviluppo di sistemi quantistici per la distribuzione di chiave è necessario fare una premessa e discutere dell'impatto che ha avuto il Cifrario di Vernam sul panorama delle comunicazioni cifrate. Questa codifica, anche nota come One Time Pad (OTP)[5], fu la prima ad essere riconosciuta come perfetta. Il livello di sicurezza che questa tecnica garantisce permette di scambiare messaggi che con certezza matematica non potranno essere compresi da nessuno anche avendo a disposizione tutto il potere computazionale che si desidera. Questo risultato però è ottenuto soltanto se vengono rispettate alcune condizioni. Il Cifrario di Vernam infatti richiede che per ogni messaggio cifrato il destinatario abbia a disposizione una chiave segreta condivisa con il mittente e che questa sia lunga quanto il messaggio stesso. La tecnica usata per cifrare il messaggio associa le lettere del messaggio con quelle della chiave in un rapporto 1:1 e usando una sommatoria modulo 26, nella stessa maniera vista nel Disco Cifrante di Leon Battista Alberti, produce la lettera del messaggio cifrato. La sicurezza incondizionata di questo codice è stata dimostrata da Shannon nel 1949[6] affermando inoltre che per la validità di questo risultato fosse necessario che la chiave venga generata in maniera casuale e completamente indipendente dal messaggio stesso.

Il risultato ottenuto dal codice di Vernam era notevole ma le condizioni richieste per la sua applicazione erano difficili da soddisfare. Una chiave casuale, indipendente dal messaggio e lunga tanto quanto questo non era facile né da generare né da gestire. Le prime realizzazioni di questa cifratura non rispettavano affatto questi vincoli. Venivano riempiti interi taccuini con la chiave, che per praticità erano scambiati e riutilizzati infrangendo in questo modo tutte le indicazioni esposte da Shannon. Le cifrature così realizzate erano estremamente vulnerabili, non potendo in nessun modo offrire lo stesso livello di sicurezza garantito a livello teorico.

Grazie allo sviluppo delle applicazioni crittografiche della fisica quantistica è ora possibile realizzare un effettivo generatore di numeri casuali[1], fondamentale per una corretta produzione della chiave della crittografia OTP, inoltre grazie al progresso della tecnologia di QKD è ora possibile garantire una sicurezza incondizionata sullo scambio della chiave tra gli utenti. Utilizzando questi sviuppi tecnologici è ora possibile una corretta implementazione della crittografia OTP che grazie alla combinazione delle tecniche di QKD e del QRNG può finalmente raggiungere il risultato teorico di crittografia perfetta.

Utilizzare la trasmissione di stati quantistici per la comunicazione della chiave permette infatti di godere dei risultati del *no-cloning theorem*. Questo teorema dimostrato da Wootters e Zurek nel 1984[7] afferma che è impossibile duplicare uno stato quantistico mantenendo inalterato l'originale. In altre parole questo teorema implica che è impossibile per un utente malevolo riuscire a carpire informazioni su una comunicazione quantistica senza rivelare la propria presenza. Questo risultato di fondamentale importanza per la sicurezza del sistema, distintivo delle comunicazioni quantistiche, permette di realizzare un canale sicuro tramite il quale gli utenti possono scambiarsi le chiavi in sicurezza.

In linea di principio esiste un'ampia scelta tra i mezzi che possono essere utili per la realizzazione di una comunicazione quantistica, infatti sono stati proposti esperimenti dove sono stati utilizzati ioni, fotoni, atomi e molecole[8]. Ad ogni modo l'unica scelta presa in considerazione per lo sviluppo della QKD è stata la luce. Infatti il processo di scambio delle chiavi assume significato soltanto se esiste una considerevole distanza macroscopica che separa gli utenti coinvolti nella comunicazione, in questo caso grazie al fatto che è molto facile spostare i fotoni ed inoltre che questi posseggono una bassa interazione con la materia, consistono dell'unica scelta pratica per la realizzazione di una comunicazione quantistica. Nei casi più comuni di applicazioni, viene fatto uso di fotoni che viaggiano su fibre ottiche[9] o di telescopi per la realizzazione di collegamenti *free space*[2][10].

Lo scenario tipico di applicazione della QKD[11] prevede che i due utenti comunemente chiamati Alice e Bob dispongano di due canali di comunica-

[1] Sfruttando la natura quantistica della luce è possibile realizzare un Quantum Random Number Generator (QRNG) semplicemente usando un polarizing beam splitter (PBS) ed un single photon detector (SPD). Polarizzando il fotone a 45° infatti esso avrà uguale probabilità di causare un rilevamento sia lungo l'asse verticale (V) sia lungo l'asse orizzontale (H), generando in questo modo un flusso casuale di bit.

[2] Oltre all'utilizzo della fibra è possibile trasmettere un segnale ottico anche in spazio aperto tramite l'uso di telescopi, questo genere di collegamenti vengo comunemente chiamati *free-space*.

zione, uno classico utilizzato per lo scambio di messaggi ausiliari alla QKD e per l'invio del messaggio cifrato, e uno quantistico dove verranno comunicati gli stati quantistici. In questo modello il canale classico è considerato un canale autenticato, il che significa che i due utenti sono in grado di riconoscere con chi stanno comunicando. In questo modo un terzo utente malevolo, comunemente chiamato Eve potrà solamente essere in grado di ascoltare la loro comunicazione lungo il canale classico e in nessun modo alterarla. Al contrario il canale quantistico, in questo modello è considerato sia ascoltabile che manipolabile da Eve. Nella pratica, il livello di sicurezza che questa tecnica garantisce è raggiunto assicurando che la chiave usata dagli utenti sia segreta. Se la chiave scambiata è considerata segreta si procede con il protocollo altrimenti la sessione viene terminata e l'intera procedura viene ripetuta. Come già accennato infatti se Eve è in ascolto sul canale quantistico la comunicazione tra Alice e Bob viene degradata a causa dell'errore introdotto da Eve, in questo modo, quantificando l'errore si è in grado di stimare la quantità di informazione percepita da Eve e scegliere se sia il caso o meno di ripetere lo scambio della chiave.

2.1 Il protocollo BB84

Nel 1984 C. Bennett e G. Brassard pubblicano il primo protocollo utilizzabile per instaurare una comunicazione basata sulla quantum key distribution, noto con il nome di BB84[12][13]. Questo protocollo prevede che Alice e Bob siano in grado di trasmettere e ricevere fotoni tramite il canale quantistico che possano essere polarizzati usando la base rettangolare (+) oppure usando la base diagonale (×). Nello specifico l'informazione viene codificata in questo modo: la polarizzazione verticale (V) e antidiagonale (A) codificano il bit 1 invece la polarizzazione orizzontale (H) e diagonale (D) codificano il bit 0. Prima che venga applicato il protocollo BB84 è necessario effettuare una fase di allineamento, dove Alice mandando una sequenza di coppie (bit, polarizzazione) nota permette a Bob di allineare le sue basi e così facendo annullare le rotazioni che la fibra ottica applica all'impulso luminoso lungo il tragitto. Il protocollo BB84 si sviluppa seguendo tre fasi:

1. Nella prima fase Alice invia a Bob un fotone tramite il canale quantistico utilizzando una a caso delle due basi di polarizzazioni descritte qui sopra. In seguito Bob lo rileva utilizzando analogamente una a caso delle due basi. Questa sequenza viene ripetuta N volte, in questo modo Bob si troverà ad avere una sequenza di N coppie (bit, base).

2. Successivamente Alice e Bob si scambiano, tramite il canale classico, la lista dei valori delle basi usate per rilevare i fotoni. In questo modo possono scartare le coppie che sono state interpretate usando basi differenti. Questo meccanismo è chiamato *sifting*. Al termine di questa procedura entrambi gli utenti condividono un insieme di bit noto come *raw key*.

3. Infine Alice e Bob condividono pubblicamente una porzione della raw key, in questo modo sono in grado di stimare gli errori introdotti dal canale quantistico e verificare se Eve di trova in ascolto. A questo punto vengono applicate tecniche classiche di *post-processing*[3], come l'error correction e la privacy amplification che permettono di rimuovere la porzione di informazione sulla chiave che possibilmente si trova in possesso di Eve. Al termine di questa fase si possono verificare due eventualità: o entrambi gli utenti condividono una chiave segreta e quindi possono procedere nella comunicazione del messaggio cifrato, oppure gli utenti hanno verificato che durante la comunicazione della chiave il tasso di errore è stato troppo elevato e che quindi troppa informazione potrebbe essere trapelata a un eventuale utente malevolo in ascolto, in questo caso il protocollo si interrompe e si ricomincia da capo.

2.2 Varianti del BB84

Con il tempo sono state proposte molte varianti del protocollo BB84 che ritoccando alcuni suoi aspetti chiave puntano a migliorarne sensibilmente le performance. Vale la pena di presentare alcune delle soluzioni che sono state trovate per approfondire gli stratagemmi usati per aumentare l'efficacia di questo protocollo conservando allo stesso tempo la sua sicurezza incondizionata.

Efficient BB84: questa variante è stata realizzata con l'obbiettivo di ridurre il tempo necessario per la generazione della chiave, andando ad ottimizzare la fase di sifting riducendo il numero di stati scartati[14]. Durante la trasmissione della chiave, seguendo le indicazioni specificate dal protocollo BB84, è possibile che si verifichi l'eventualità in cui la base scelta da Bob per interpretare l'impulso inviato da Alice risulti essere sbagliata, in questo caso l'impulso viene scartato aumentando il tempo necessario per ottenere

[3]Il *post-processing* è una fase dell'esecuzione del protocollo che avviene successivamente l'elaborazione dei dati. In questa fase è possibile effettuare tutte le correzioni utili ad eliminare dai risultati ottenuti una porzione degli errori identificati.

la lunghezza desiderata della chiave sicura. L'unico accorgimento applicato da questa variante consiste nello sbilanciare le probabilità di scelta delle basi usate per la generazione degli impulsi luminosi, rendendo quindi più rara la necessità di scartare impulsi. Usando la base (+) per la trasmissione della chiave e la base (×) per verificare la presenza di Eve si riesce a velocizzare l'intero processo senza rinunciare alla sicurezza incondizionata che offre il protocollo.

Three-state protocol: per semplificare l'implementazione del protocollo efficient BB84, sia lato Alice che lato Bob, è possibile utilizzare nella trasmissione soltanto tre dei quattro stati ($[V, H], [D, A]$) usati canonicamente[15]. In questo modo, scartando uno dei due stati dalla base di controllo è possibile abbassare la complessità del sistema senza perdere le garanzie del protocollo.

B92: questo protocollo venne proposto da Bennett nel 1992[16] usando una configurazione semplificata rispetto a quella descritta per il BB84, andando in questo modo a ridurre la complessità del sistema necessario per la trasmissione. In questo protocollo vengono trasmessi lungo il canale quantistico soltanto gli stati ([H],[D]), così facendo il no-cloning theorem continua a valere. Lato ricevitore invece si continuano ad usare entrambe le basi (+, ×) per decodificare l'impulso luminoso. La particolarità di questo protocollo consiste nel fatto che se Alice trasmette lo stato (H) e Bob sbaglia base di interpretazione e usa (×) a questo punto entrambi i risultati (D, A) hanno ugual probabilità di manifestarsi, e in entrambi i casi il protocollo BB84 dovrebbe scartare l'impulso, nel caso del protocollo B92 invece soltanto il caso (D) significherebbe un errore perché nel caso l'esito risultasse essere (A) siccome vengono trasmessi soltanto gli stati (H, D) lo stato trasmesso non potrebbe che trattarsi di (H). La stessa osservazione vale nel caso lo stato trasmesso sia (D) e per decodificarlo venga utilizzata la base sbagliata (+). Malgrado questo vantaggio di praticità, il protocollo B92 è stato dimostrato essere vulnerabile agli attacchi di Eve, infatti se quest'ultima mettesse in atto un attacco *man in the middle* intercettando gli impulsi luminosi e inoltrandoli a Bob soltanto in caso di corretta decodifica, questo aumenterebbe il tasso di perdita del segnale facendo rimanere inalterato il QBER e riuscendo in questo modo a celare la propria presenza.

2.3 Protocolli entanglement-based

Le tecniche di quantum key distribution (QKD) possono essere applicate anche in scenari differenti. Finora abbiamo trattato il caso in cui è Alice a preparare il segnale quantistico e a condividerlo con Bob, questa configurazione è nota in letteratura con il termine di *Prepare & Measure* (P&M) ma

non è la sola disponibile. Soluzioni alternative prevedono che sia un terzo dispositivo a fornire gli stati quantistici ad Alice e Bob, come è possibile osservare in Fig.[2.1]. Seguendo questo schema è infatti possibile sfruttare le proprietà dell'entanglement quantistico con cui è possibile produrre coppie di impulsi luminosi fortemente correlati che quindi possono essere osservate indipendentemente da Alice e Bob e portare alla stessa misurazione. I vantaggi di questo approccio riguardano l'implementazione del sistema, delegando la generazione degli impulsi casuali ad un terzo dispositivo l'architettura di Alice e Bob diventa simmetrica semplificando la propria realizzazione. Dal punto di vista della sicurezza invece continuano a valere i risultati presentati per le architetture P&M, inoltre siccome nessuna informazione viene codificata e trasmessa con gli stati quantistici non è necessario riporre fiducia nella sorgente degli impulsi luminosi, in quanto non apre nessuna vulnerabilità del sistema. Anche per questa configurazione Bennett e Bressard insieme a N. D. Mermin hanno proposto un protocollo molto simile al BB84 chiamato BBM92[17], dove allo stesso modo visto in precedenza vengono usate entrambe le basi (+, ×) per la trasmissione e la ricezione degli impulsi luminosi, in questo modo se Eve tentasse di intromettersi nella comunicazione la sua presenza verrebbe rilevata grazie all'inevitabile incremento nel QBER che causerebbe.

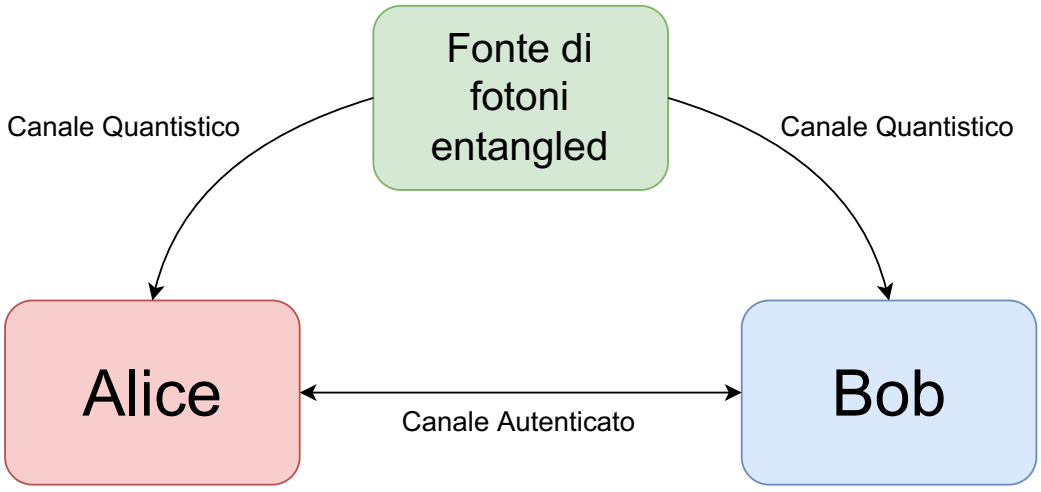


Figura 2.1: Schema di comunicazione per protocolli di QKD entanglement-based

2.4 Attacchi alla QKD

Anche se alcune delle tecniche di QKD sono state dimostrate essere in grado di garantire una sicurezza incondizionata è necessario prestare delle attenzioni. Infatti se questa tecnologia viene applicata senza prestare le giuste accortezze può rendersi vulnerabile ad alcune categorie di attacchi da cui dovrebbe essere protetta. Di seguito vengono riportati due esempi di attacchi in grado di violare implementazioni erronee di sistemi QKD.

2.4.1 Intercept and Resend Attack

Questo tipo di attacco prevede che Eve sia in grado di intromettersi nella comunicazione che Alice sta avendo con Bob tramite il canale quantistico ed intercettare gli impulsi luminosi che quest'ultima manda a Bob in modo da ricavare la chiave che i due utenti stanno cercando di condividere. Inoltre, per nascondere la propria presenza e far procedere ignara la comunicazione, Eve trasmette una copia degli impulsi luminosi anche a Bob, da qui il nome intercept and resend attack. Questa situazione non dovrebbe potersi verificare, infatti come abbiamo visto quando è stato introdotto il no-cloning theorem se Eve cerca di intromettersi ed intercettare la comunicazione tra Alice e Bob allo stesso tempo non può fare a meno di alzare il livello di errore quantistico (QBER) e rivelare la propria presenza. Se però alla modulazione ottica usata, ossia quella della base rettangolare (+) in combinazione a quella diagonale (×), sostituiamo una più comune ed economica modulazione OOK[4] questo non resta più vero. L'utilizzo della combinazione delle basi per la codifica del segnale quantistico infatti non è una scelta ingiustificata. L'implementazione di tecniche di modulazione ottica più elementari infatti non sono sufficienti per far valere le condizioni del no-cloning theorem che richiedono che il segnale quantistico non debba essere rappresentato da stati ortogonali.

2.4.2 Photon Number Splitting Attack

Il passaggio dal modello teorico a quello pratico non è mai semplice, molto spesso si rende necessario scendere a compromessi che ne rendano possibile la realizzazione evitando di allontanarsi troppo da quella che era l'idea originale. Questo è il caso del generatore di impulsi luminosi coinvolto nella

[4]Definita con il termine on-off keying (OOK) questa modulazione binaria è in assoluto la più semplice, essa associa un impulso luminoso al bit con valore 1 mentre assegna l'assenza di impulsi luminosi ai bit 0. In questo modo è possibile effettuare una trasformazione diretta dal flusso informativo di bit a quello luminoso immesso nel canale di trasmissione.

QKD, infatti anche se questo viene descritto come capace di produrre impulsi a singolo fotone nella pratica vengono utilizzati dei laser attenuati capaci di produrre impulsi di luce coerente che soltanto statisticamente sono composti da un singolo fotone. Usando questi strumenti infatti non è possibile garantire che gli impulsi siano composti da singoli fotoni andando così in contrasto con le indicazioni del modello teorico che ne garantivano la sicurezza. Sfruttando questa mancanza nell'implementazione è possibile mettere in atto un attacco per l'intercettazione della chiave comunemente chiamato Photon Number Splitting (PBS). Usando una speciale tecnica di misurazione in grado di dividere l'impulso ottico Eve è in grado di inoltrare a Bob l'impulso di Alice mantenendone allo stesso tempo una porzione per sé. Il risultato di questo procedimento è che Eve guadagna la stessa informazione che ha Bob sulla chiave segreta riuscendo a celare la propria presenza agli utenti coinvolti nella comunicazione. Allo scopo di impedire il verificarsi di questo scenario è stata proposta un'aggiunta al protocollo BB84 consistente nella tecnica *decoy-state*. Questa tecnica prevede che Alice trasmetta la chiave a Bob usando diversi livelli di intensità luminosa seguendo una statistica nota in partenza. Successivamente in fase di sifting insieme alle basi con cui è stata interpretata la chiave, Alice inoltra anche le intensità scelte per gli impulsi luminosi inviati, in questo modo Bob può dedurre la quantità di impulsi composti da un singolo fotone e quantificare l'informazione trapelata che ha raggiunto Eve.

L'aggiunta della tecnica di decoy-state anche se non prevista nella teoria del protocollo BB84 ne ha facilitato l'implementazione agevolandone la diffusione. La possibilità di utilizzare laser ad impulsi attenuati al posto di Quantum Dots[5] con impulsi a singolo fotone ne ha semplificato l'implementazione, permettendo di usare strumenti più comuni e meno elaborati per la realizzazione del protocollo.

2.5 L'implementazione del gruppo QuantumFuture

L'implementazione della QKD scelta per laboratori di Padova da parte del gruppo QuantumFuture prevede uno schema P&M facente uso del protocollo BB84 Three state con l'aggiunta della tecnica di decoy-state.

[5]Gli emettitori Quantum Dots sono strumenti in grado di produrre in maniera controllata impulsi composti da un singolo fotone, ma essendo più costosi e sofisticati dei più comuni laser il loro utilizzo rimane limitato.

Per poter raggiungere questo risultato il gruppo QuantumFuture ha assemblato i due sistemi, rispettivamente Alice e Bob, dotandoli ognuno degli strumenti necessari per portare a termine la comunicazione. Com'è possibile intuire dalla Fig.2.2 i dispositivi necessari sono parecchi. Per fare chiarezza e rendere più comprensibile la routine necessaria per istanziare una comunicazione tra Alice e Bob, vengono di seguito affrontati singolarmente i dispositivi coinvolti in ordine di utilizzo temporale lungo la pipeline del protocollo.

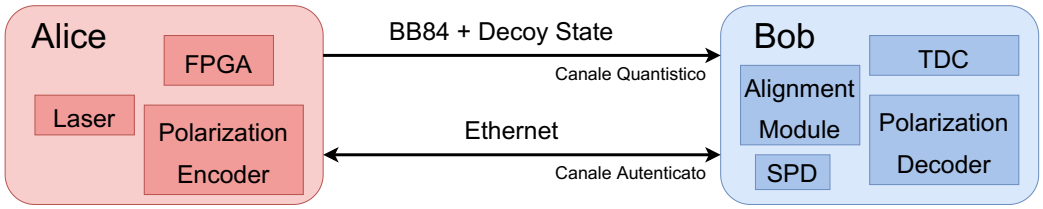


Figura 2.2: Rappresentazione dello schema di comunicazione tra Alice e Bob, mettendo in evidenza i principali componenti coinvolti.

Field Programmable Gate Array (FPGA): consiste in una scheda elettronica programmabile capace di produrre in output segnali dotati di specifiche proprietà e relazioni. Nello sviluppo del progetto per la QKD questa scheda è stata usata per produrre i segnali elettrici necessari al funzionamento di Alice ed in generale a quello della QKD. Alcuni di questi segnali sono: il clock di riferimento di Alice, il segnale necessario per la stimolazione del laser ed il segnale di riferimento per Bob nella variante ExtRef del protocollo di sincronia. Tutti questi segnali devono essere strettamente legati tra loro ed è quindi necessario che vengano prodotti da un'unica sorgente, per questo motivo si è scelto di delegare la generazione di questi segnali ad un singolo componente. Per ottenere segnali sufficientemente precisi si è scelto di limitare il clock di Alice per la stimolazione del laser a $50MHz$.

Laser: questo strumento, ormai indispensabile per le comunicazioni su fibra ottica, permette di produrre impulsi di luce coerente tramite il processo di emissione stimolata. Le lunghezze d'onda che sono state utilizzate per realizzare la QKD sono $1310nm$ e $1550nm$ in quanto sono quelle che vengono meno attenuate dalla trasmissione su fibra ottica.

Polarization Encoder: è un dispositivo tramite il quale è possibile manipolare la polarizzazione degli impulsi luminosi riuscendo a riprodurre tutte le polarizzazioni necessarie alla QKD. Nello specifico il polarization encoder usato in questo progetto (POGNAC) è stato ideato e sviluppato dal gruppo

QuantumFuture [18] e fa uso di un modulatore di fase Mach-Zehnder con un cristallo di $LiNbO_3$ pilotato da un segnale RF inserito all'interno di un interferometro di Sagnac. La scelta di realizzare un nuovo polarization encoder è stata presa con l'obbiettivo di proporre una soluzione alternativa formata da componenti semplici ed economici ma che allo stesso tempo fosse in grado di compensare errori dovuti a bias e drift termici. Il funzionamento del dispositivo prevede che il laser produca impulsi con polarizzazione nota [V] che entrando in ingresso al circolatore ottico (CIRC) nella porta 1 verranno indirizzati alla porta 2. Successivamente gli impulsi attraverseranno il polarization controller (PC) che modificherà la loro fase facendola passare da [V] ad una sovrapposizione bilanciata degli stati ([H],[V]). Gli impulsi luminosi in seguito verranno dati in ingresso al interferometro Sagnac all'interno del quale è presente il modulatore di fase. Grazie al fatto che ad entrambi gli impulsi luminosi, sia quello con polarizzazione [V] che quello con polarizzazione [H], viene fatto fare lo stesso percorso soltanto in direzione opposta la polarizzazione dell'impulso risultante dipenderà soltanto dalla differenza di fase delle due componenti, risultando così immune ad altri errori dovuti alla componentistica.

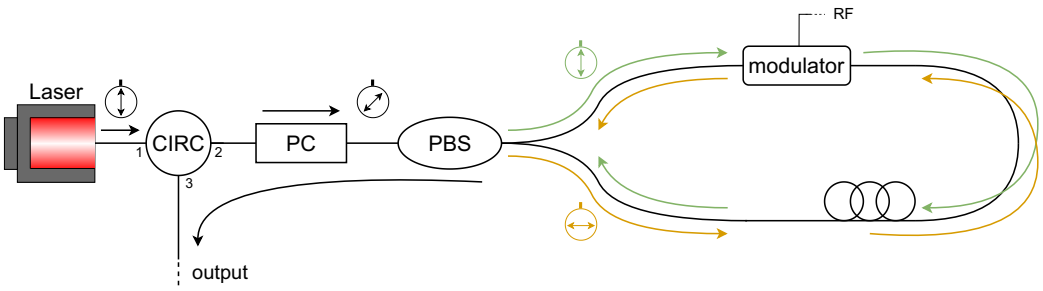


Figura 2.3: Rappresentazione del polarization encoder POGNAC realizzato dal gruppo QuantumFuture

Polarization Decoder: è un dispositivo necessario per Bob per riconoscere la polarizzazione degli impulsi ottici inviati da Alice e trasferire quest'informazione quantistica dalla fase alla posizione temporale dell'impulso. La soluzione utilizzata dal gruppo QuantumFuture prevede che l'informazione codificata nella polarizzazione degli impulsi ottici, una volta raggiunto Bob, venga trasferita realizzando un intervallo temporale composto da 4 slot che con l'utilizzo della tecnica di time multiplexing possa essere utilizzato per semplificare l'elaborazione dei dati. Come è possibile osservare in Fig.2.4 gli impulsi ottici ricevuti da Alice entrano in input ad un beam splitter (BS) che in maniera casuale inoltra il fotone ad uno dei due successivi polarization

beam splitter (PBS) che sono in grado di rilevare correttamente rispettivamente la base rettangolare ([H],[V]) e la base diagonale ([D],[A]). Arrivati a questo punto, in base a quale delle 4 possibili polarizzazioni è stata rilevata, gli impulsi vengono ritardati rispettivamente di $0, \Delta t, 2\Delta t, 3\Delta t$ in questo modo è possibile fare in modo che ciascuno degli impulsi vada a riempire lo slot ad esso dedicato all'interno dell'intervallo temporale. Per ragioni che verranno trattate in seguito, si è scelto di trasmettere fotoni con una frequenza di $50MHz$ in questo modo gli impulsi ottici vengono sparati uno ogni $20ns$ e con la stessa frequenza vengono ricevuti da Bob. Partizionando l'intervallo di $20ns$ in 4 slot ciascuno di $5ns$ è possibile utilizzare lo schema di time multiplexing scelto dal gruppo QuantumFuture. Quello che accade all'interno del polarization decoder consiste nell'applicare un ritardo controllato in base alla polarizzazione dell'impulso luminoso ed in questo modo farlo rientrare nello slot appropriato.

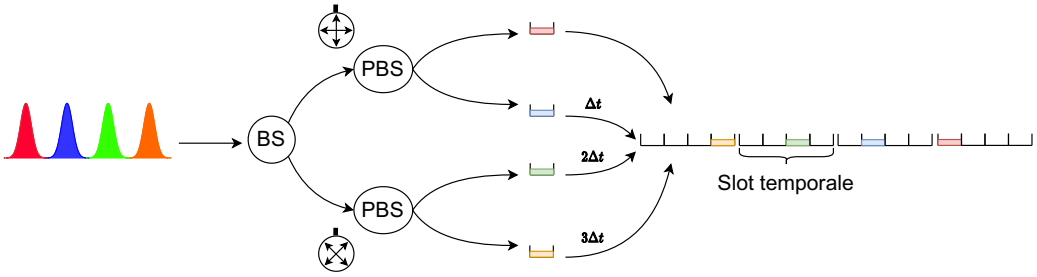


Figura 2.4: Rappresentazione del polarization decoder

Automatic Polarization Controller (APC): è un dispositivo che permette di compensare le variazioni nella polarizzazione degli impulsi luminosi che vengono trasmessi. La fibra ottica, lungo il suo tragitto, può applicare delle rotazioni alla polarizzazione degli impulsi luminosi. Per evitare letture erronee è quindi necessario che queste variazioni vengano compensate e che si ripristini la corrispondenza tra le polarizzazioni degli impulsi inviati da Alice e quelli ricevuti da Bob. L'APC è un dispositivo che applicando una serie di trasformazioni meccaniche alla fibra ottica è in grado di ripristinare la correlazione tra gli impulsi inviati e quelli ricevuti. Quello che accade nella pratica è che prima di iniziare il protocollo BB84 viene eseguita una fase di allineamento dove Alice inviando una sequenza di stati quantistici noti permette a Bob, grazie appunto a questo strumento, di far corrispondere gli stati.

Single Photon Detector (SPD): è responsabile della trasformazione del segnale quantistico che passa dall'essere ottico all'essere di natura elettrica. Questa conversione è essenziale per poter elaborare i dati relativi gli istanti di arrivo del segnale quantistico. Come abbiamo visto nei precedenti paragrafi, il problema di identificare la polarizzazione usata è mutato, e tramite la tecnica di time multiplexing è diventando quello di dover distinguere quale dei quattro slot in cui è stato diviso l'intervallo temporale ospita l'impulso che ora è diventato elettrico. Data l'elevata sensibilità di questo strumento, che ripetiamo deve poter rilevare l'arrivo di singoli fotoni, successivamente ad una rilevazione è necessario un tempo di hold-off di $20\mu s$ in cui lo strumento si mostra refrattario agli arrivi di ulteriori impulsi luminosi. Questo periodo di tempo permette di eliminare l'effetto delle correnti parassite che intaccano il funzionamento dei componenti elettronici, nello specifico avviene il rilassamento degli elettroni caduti negli stati trappola della zona di svuotamento delle giunzioni pn, che altrimenti potrebbero dare luogo a false rilevazioni incrementando il livello di rumore. Questo meccanismo di sicurezza fondamentale ha come conseguenza una saturazione del detector, che non può produrre più di 50000 segnali al secondo. Questa saturazione contribuisce all'attenuazione del segnale quantistico, che pur essendo emesso da Alice con una frequenza di $50MHz$, arriva a Bob limitato al massimo a $50kHz$ a causa del tempo di hold-off.

Time Digital Converter (TDC): è uno strumento fondamentale per il funzionamento della QKD. Il suo funzionamento permette di fornire un riferimento temporale tramite il quale elaborare tutti i segnali dati in ingresso a Bob. La versione utilizzata dal gruppo QuantumFuture offre la possibilità di avere multipli canali di ingresso e per ognuno offre un canale output dove vengono ritornati i timestamps relativi al segnale fornito come input. Il TDC che è stato adottato è fornito di un clock interno che gli permette di acquisire misurazioni con precisione digitale di $81ps$ circa, riuscendo in questo modo ad abbinare agli impulsi elettrici inoltrati dal SPD, un numero rappresentante la quantità di periodi del clock del TDC trascorsi da quando è stato acceso a quando è stata osservata la misura. I dati forniti dal TDC, oltre a essere quelli che verranno effettivamente elaborati per l'estrazione della chiave, sono anche usati dai protocolli di sincronia come il Qubit4Sync e l'ExtRef. Da questi dati infatti è possibile risalire a tutti i parametri fondamentali per stabilire la sincronia nella comunicazione tra Alice e Bob.

Capitolo 3

Protocolli di sincronia

Per poter instaurare una comunicazione tra due dispositivi è necessario che questi siano sincronizzati. Lo stesso vale per Alice e Bob, negli istanti in cui Alice invia gli impulsi luminosi a Bob quest'ultimo per riceverli correttamente deve essere sincronizzato con Alice, riuscendo in questo modo ad interpretare senza errori il segnale trasmesso sul canale quantistico.

Per due dispositivi essere sincronizzati significa avere lo stesso riferimento temporale e muoversi coerentemente rispetto ad esso. Lo scenario presentato nei capitoli precedenti mette in evidenza la necessità di un sistema di sincronizzazione particolarmente sofisticato. Infatti se per la maggior parte delle applicazioni un livello di precisione di un millisecondo è più che sufficiente, come avremo modo di osservare per l'applicazione della QKD, siccome l'impulso ottico deve essere centrato all'interno di un intervallo di un nanosecondo, sono necessarie tecniche più elaborate che alzando notevolmente la complessità del sistema spingono verso soluzioni non canoniche.

Un protocollo di sincronia consiste in un insieme di procedure utili a ricavare i parametri necessari ad instaurare la sincronia tra due o più dispositivi, attivando inoltre meccanismi per il mantenimento di questo stato al fine di permettere lo svolgersi della comunicazione. Per fare in modo che Bob si sincronizzi con il segnale emesso da Alice è necessario ricavare il *periodo* e la *fase* con cui quest'ultimo arriva al ricevitore. Grazie all'utilizzo del TDC è possibile avere un riferimento temporale su cui impostare le nostre elaborazioni, basato sulle etichette temporali che questo dispositivo abbina ad ogni impulso ricevuto. Le calibrazioni che in seguito vengono applicate, per permettere lo scambio di informazioni tra Alice e Bob, fanno parte del *pre-processing*[1] e consistono tutte in aggiustamenti che vanno a modificare i riferimenti del

[1] Il *pre-processing* è una fase dell'esecuzione del protocollo che precede l'elaborazione dei dati. In questa fase è possibile effettuare tutte le correzioni necessarie a facilitare l'elaborazione e l'ottenimento dei risultati.

software, lasciando inalterato il funzionamento hardware delle apparecchiature. Avendo la necessità di rilevare impulsi a singolo fotone serve utilizzare strumenti estremamente sensibili, se in altre applicazioni il segnale è chiaramente distinguibile dal rumore di fondo grazie alla sua intensità e potenza lo stesso non si può affermare in questo caso, se non vengono messi in atto i giusti accorgimenti risulta difficile persino riuscire a distinguere il segnale informativo.

Entrambi i dispositivi, Alice e Bob, sono dotati di un clock interno ma essendo indipendenti non hanno motivo di battere lo stesso periodo. Per questo motivo per riuscire a identificare le posizioni di arrivo degli impulsi ottici inviati da Alice è necessario misurare il periodo del segnale emesso da quest'ultima utilizzando il sistema di riferimento di Bob. Una volta determinato il periodo di arrivo degli impulsi è possibile definire la *Time Window*, ossia la finestra temporale dove è previsto l'arrivo di un qubit. Nella configurazione adottata per la realizzazione della QKD da parte del gruppo QuantumFuture, va ricordato che la frequenza con cui Alice trasmette il segnale quantistico è di $50MHz$ e che quindi è presente un impulso luminoso ogni $20ns$. All'uscita del polarization decoder utilizzato da Bob però a causa del time multiplexing l'impulso viene costretto ad occupare uno dei 4 slot da $5ns$, trasformando il segnale dotandolo di una forte componente di Fourier di periodo $5ns$.

Come è possibile osservare in Fig.[3.1], all'interno di uno slot temporale, deve venire posizionato l'intervallo di campionamento, ossia l'effettivo intervallo temporale dove verrà osservato il segnale per verificare o meno la presenza di un impulso. Questo intervallo temporale ha ampiezza $1ns$ e se l'impulso dovesse trovarsi al di fuori di esso sarebbe scartato e l'informazione codificata verrebbe persa. Riuscire a conoscere in anticipo l'istante in cui si aspetta l'arrivo di un impulso ci permette di scartare la maggior parte delle rilevazioni dovute a rumore ed in questo modo aumentare il signal-to-noise ratio (SNR[2]), uno dei fattori principali per determinare le performance di una comunicazione.

Una volta ricavata l'informazione sul periodo del segnale è necessario calcolare la fase che questo ha nei confronti del sistema di riferimento temporale di Bob. Con il termine fase in questo contesto ci si riferisce allo shift temporale che è necessario applicare al segnale quantistico per fare in modo che quest'ultimo si trovi ad essere centrato rispetto alla Time Window.

[2]Il *Signal Noise Ratio (SNR)* è un indicatore di quanto forte è il rapporto tra la potenza del segnale informativo e il rumore di fondo in una comunicazione. Questo parametro è uno dei parametri fondamentali in grado di riassumere la qualità di una trasmissione.

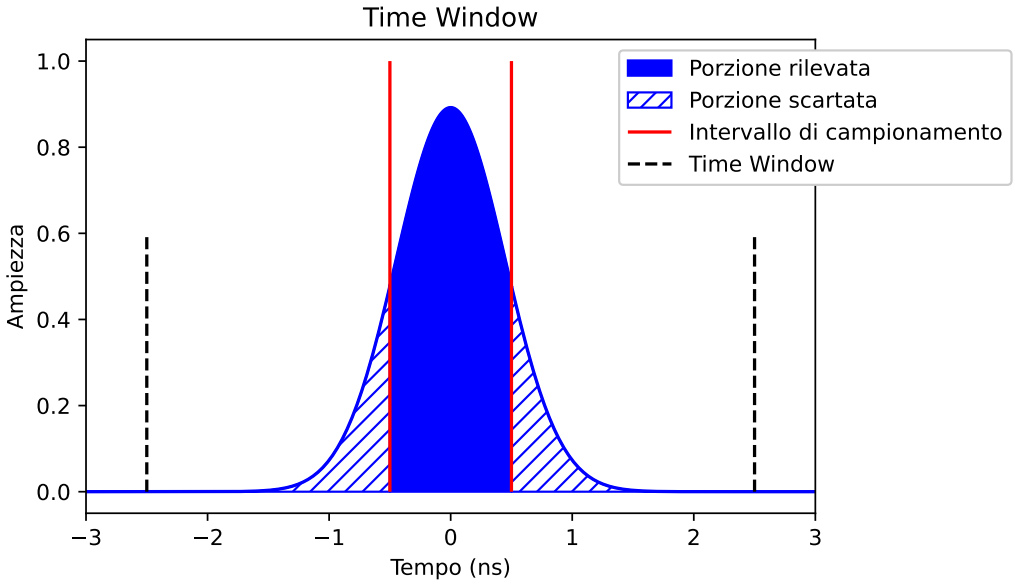


Figura 3.1: Rappresentazione della Time Window

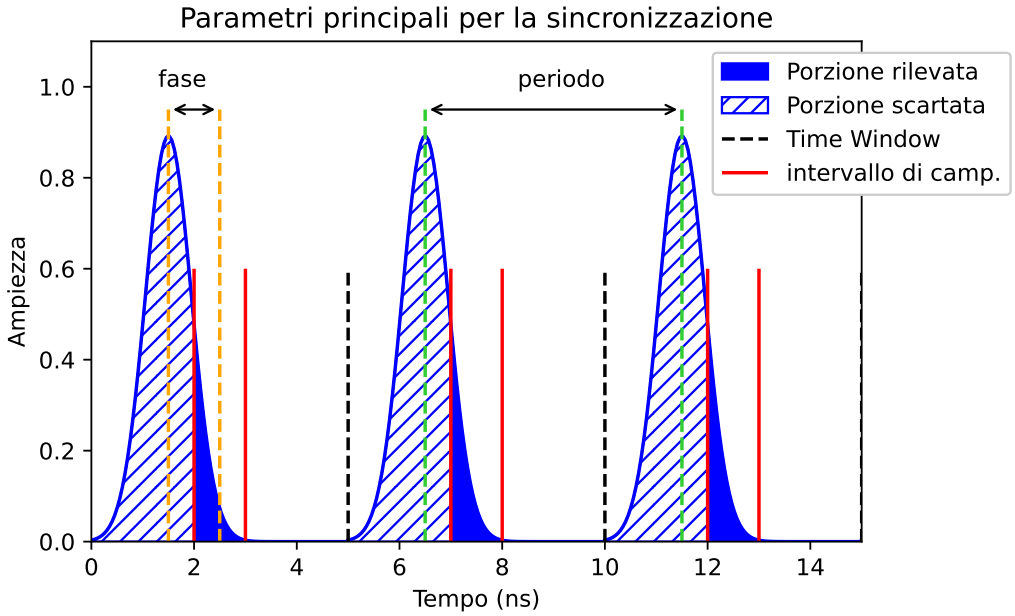


Figura 3.2: Rappresentazione dei parametri necessari per la sincronia

Com'è possibile osservare in Fig.[3.2], se non venisse applicata nessuna correzione della fase all'impulso questo potrebbe trovarsi al di fuori dell'intervallo di campionamento ed in questo modo causare un drastico calo delle performance della comunicazione.

Riuscire nell'intento di ricavare buone stime di questi parametri non è per nulla banale, una volta arrivati alla realizzazione effettiva del protocollo ci si rende conto di molte complicazioni. A seguito delle attenuazioni applicate al segnale quantistico, siano esse dovute alla saturazione del SPD causata dal tempo di hold-off oppure dovute dalla lunghezza del collegamento in fibra ottica, meno di un impulso ogni mille raggiunge effettivamente Bob. Inoltre a causa dei *drift*[3] dei clock dei vari dispositivi coinvolti nella comunicazione la fase del segnale non rimane costante oscillando in maniera imprevedibile.

Riuscire a sincronizzare Alice e Bob nello scenario appena descritto comporta la risoluzione di tutte le problematiche presentate. A questo proposito nelle sezioni successive, vengono introdotti i due protocolli di sincronia usati per la realizzazione della QKD.

[3]Il termine *drift* riferito ad un clock sta ad indicare le variazioni casuali di contrazione e dilatazione del periodo del segnale da essi fornito.

3.1 Il protocollo Qubit4Sync

Il protocollo Qubit4Sync è un metodo di sincronizzazione proposto dal gruppo QuantumFuture dei laboratori di Padova per l'applicazione della QKD[19]. Questa tecnica sfrutta gli stessi *qubits*[4] usati per trasmettere la chiave grezza allo scopo di sincronizzare i due dispositivi. Non avendo necessità di risorse aggiuntive si rivela essere una tecnica comoda e molto flessibile da applicare. Questa sua peculiarità però gli impone di operare nelle peggiori condizioni possibili. Rinunciare ad avere un riferimento esterno non significa soltanto una riduzione dei costi ed una semplificazione della progettazione del sistema, significa anche che per poter sincronizzare Alice e Bob non si potrà fare affidamento su di un segnale forte e stabile ma che si dovrà in qualche modo raggiungere il medesimo obbiettivo soltanto basandosi sui dati estratti dal flusso di qubit che arrivano al ricevitore. Il segnale quantistico infatti è estremamente debole e affetto da perdite che abbassano ulteriormente il valore del SNR. Per questi motivi, anche se in principio il segnale quantistico conservava una sua periodicità, nel momento in cui questo arriva a Bob, questa non appare più così evidente. Per poter estrapolare delle buone stime dei parametri necessari alla sincronia in queste condizioni, il protocollo Qubit4Sync applica un'approfondita analisi del segnale quantistico. Facendo riferimento alla Fig.[3.3], dove viene riportato uno schema raffigurante tutti gli step affrontati durante l'elaborazione del protocollo, vengono in seguito illustrati nel dettaglio le procedure e i meccanismi attuati in ogni singolo passaggio.

Raccolta dei dati del segnale quantistico: Prima di tutto è necessario che i timestamps registrati dal TDC, rappresentanti gli istanti di arrivo degli impulsi ottici, vengano raccolti in pacchetti e che ad intervalli regolari vengano inoltrati a Bob per l'elaborazione. Per ottenere una gestione delle risorse ottimale è stato scelto di sviluppare la QKD usando la libreria di programmazione Qt, in grado di sfruttare a pieno l'efficienza del C++ disponendo in aggiunta di varie interfacce di accesso aggiuntive che permettono di relazionarsi con i vari dispositivi coinvolti nel protocollo per orchestrare il trasferimento e l'elaborazione dei dati. Quest'aspetto del progetto verrà affrontato in maniera più dettagliata nell'Appendice A.

[4]Il termine *qubit* è usato per indicare l'unità informativa ricavata da una comunicazione quantistica, rispettivamente il quantum-bit.

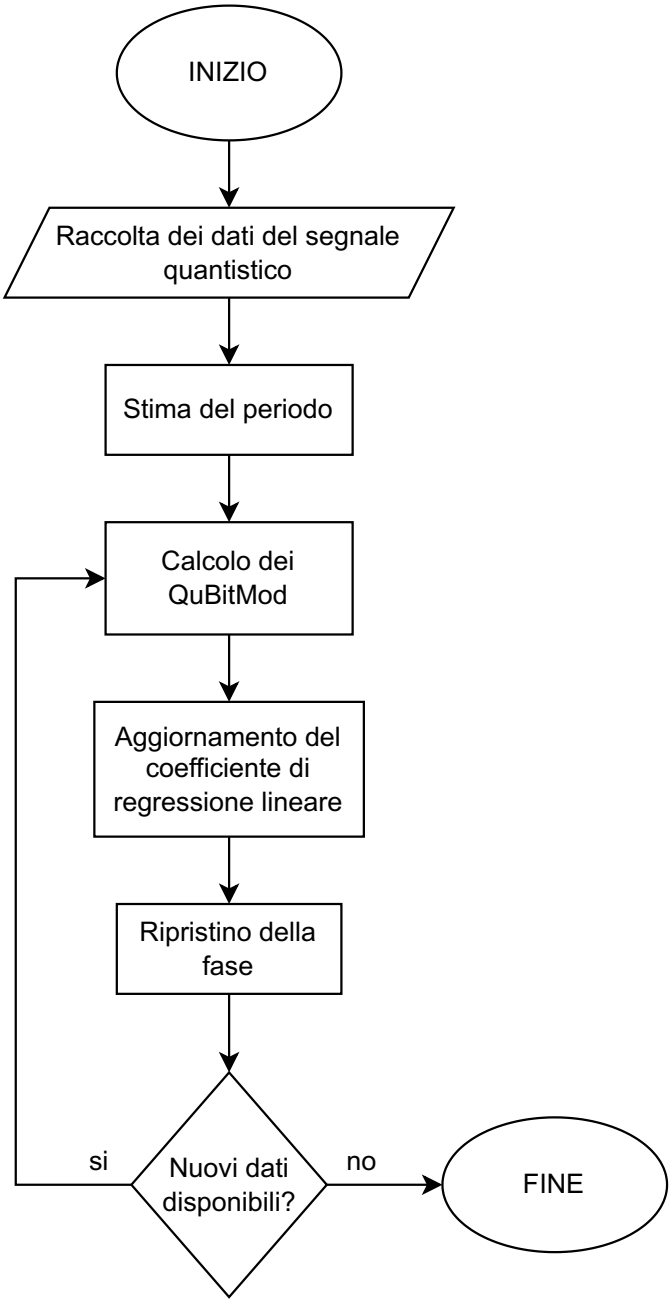


Figura 3.3: Diagramma a blocchi del funzionamento del protocollo QuBit4Sync

Stima del periodo: La stima del periodo del segnale quantistico da parte di Bob τ_0^b, è ricavata effettuando la Discrete Fourier Transform (DFT) del segnale, riportata nella formula (3.1). Usando la sequenza dei timestamps rilevata dal TDC $t_0, t_1, t_2, \ldots, t_n$ è possibile ricostruire il segnale nella sua interezza. Per riuscire ad essere elaborato in real time, data l'elevata risoluzione del segnale di $\sim 81ps$, quest'ultimo deve venire campionato in modo da ottenere $x_0, x_1, x_2, \ldots, x_N$ con $N = 10^6$ campioni.

$$X_q = \mathcal{F}_d(x_n) = \sum_{k=0}^{N-1} x_k e^{-i\frac{2\pi}{N}kq} \qquad q = 0, 1, \ldots, N-1 \tag{3.1}$$

Dal momento che si conosce l'intervallo temporale τ^a che Alice interpone tra un impulso e il successivo, ci si aspetta che il segnale quantistico abbia una frequenza nell'intorno di $1/\tau^a$. La frequenza di campionamento f_c con cui viene campionato il segnale quantistico deve quindi essere di almeno $2/\tau^a$. La DFT permette di scomporre un segnale usando una base composta da funzioni sinusoidali, alle quali sono associati i coefficienti X_k, come è possibile osservare nella formula (3.2). Questi coefficienti ci permettono di indicare quale componente frequenziale sia maggiormente presente nel segnale di partenza, ed in questo modo, anche se nel segnale ricevuto da Bob sono presenti pesanti perdite e rumore, identificare un possibile candidato per la stima di τ^b.

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i\frac{2\pi}{N}kn} \qquad n = 0, \ldots, N-1 \tag{3.2}$$

La complessità computazionale richiesta per l'applicazione di quest'analisi del segnale è $O(N^2)$, ma grazie all'utilizzo di algoritmi noti come *Fast Fourier Transform (FFT)* è possibile abbassare il costo di quest'operazione a $O(Nlog(N))$ e renderlo maggiormente efficiente. Nella Fig.[3.4] viene riportato come esempio il grafico della FFT usando il segnale quantistico registrato da Bob in un intervallo di ~ 1 secondo. Com'è possibile osservare il massimo[5] è posizionato sul valore di $50MHz$ indicando correttamente la frequenza con cui gli impulsi vengono inviati da Alice.

[5]Il picco presente al valore di frequenza $0Hz$ non è da considerarsi, in quanto non rappresenta nessuna componente periodica bensì sia un indicatore della potenza del segnale.

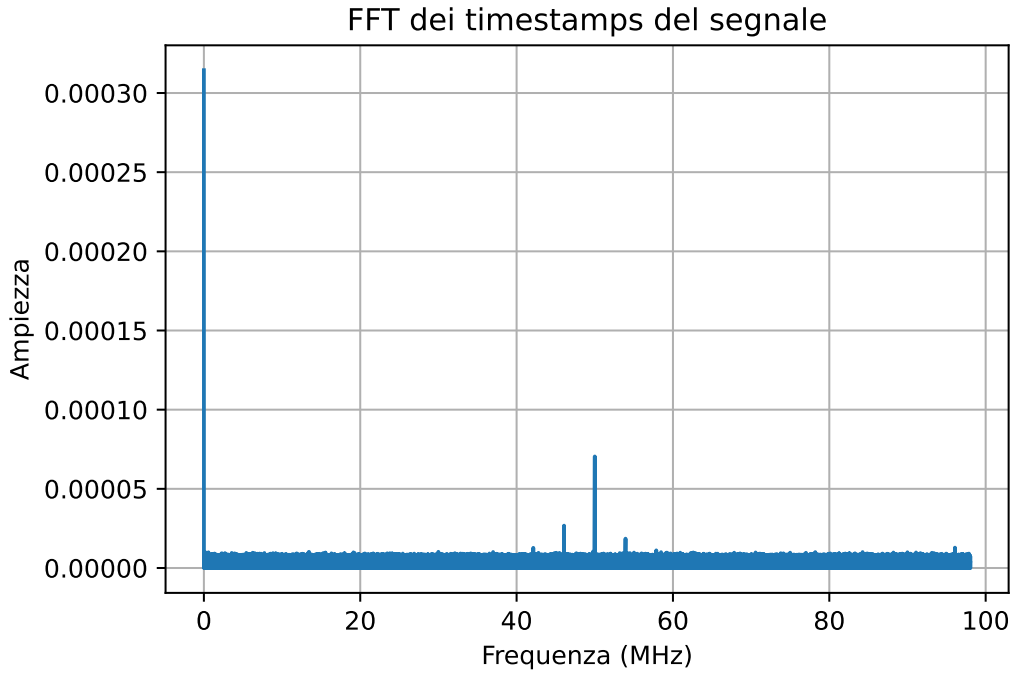


Figura 3.4: FFT del segnale quantistico ricevuto da Bob.

Calcolo dei QuBitMod: Una volta ottenuta la stima del periodo del segnale quantistico τ_0^b è possibile procedere alla computazione della sequenza dei qubitmod. Per ricavare i quibitmod è necessario partizionare la sequenza dei timestamps del segnale quantistico in slot temporali di durata uguale al valore della Time Window e osservare in che posizione interna agli slot vengono a distribuirsi gli impulsi luminosi. La distribuzione delle posizioni interne alla Time Window in cui vengono a collocarsi gli impulsi rappresenta l'informazione contenuta nei qubitmod. Per ricavare quest'informazione è necessario applicare la formula (3.3) alla sequenza t_n dei timestamps del segnale quantistico.

$$q_n = mod_{\tau_0^b}(t_n - t_0) \qquad n = 0, \ldots, N-1 \tag{3.3}$$

I qubitmod sono molto utili per osservare come sono distribuiti gli arrivi del segnale quantistico all'interno della Time Window. Per graficare quest'informazione si fa uso di un istogramma composto da tante colonne quanti sono i possibili istanti di arrivo all'interno della Time Window, più una colonna è alta più impulsi luminosi sono arrivati in quella posizione. Com'è possibile osservare in Fig.[3.5], i qubit non arrivano tutti nello stesso istante ma si distribuiscono all'interno della Time Window seguendo una distribu-

zione gaussiana. Questo fenomeno si verifica come risultato di tutti processi che disturbano la trasmissione del segnale quantistico, come il drift del clock di Alice e il drift del clock del TDC. La statistica prodotta dai qubitmod fornisce un'ottima stima di quelle che saranno le performance del protocollo di sincronizzazione che si sta usando. Idealmente vorremmo essere in grado di posizionare tutti gli impulsi del segnale al centro della Time Window, in questo modo azzereremmo la possibilità di scartare impulsi del segnale e saremmo in grado di predire con esattezza l'arrivo degli impulsi successivi. Purtroppo, a causa dei fattori disturbanti sopracitati, questo non è possibile, e gli arrivi si distribuiscono seguendo una statistica con una media e una varianza. Questi valori ci forniscono informazioni riguardo la precisione della sincronia raggiunta, infatti più sarà alto il valore della varianza più la statistica distribuirà omogeneamente gli impulsi nell'intervallo temporale causando una perdita di segnale sempre maggiore e abbassando le performance della QKD.

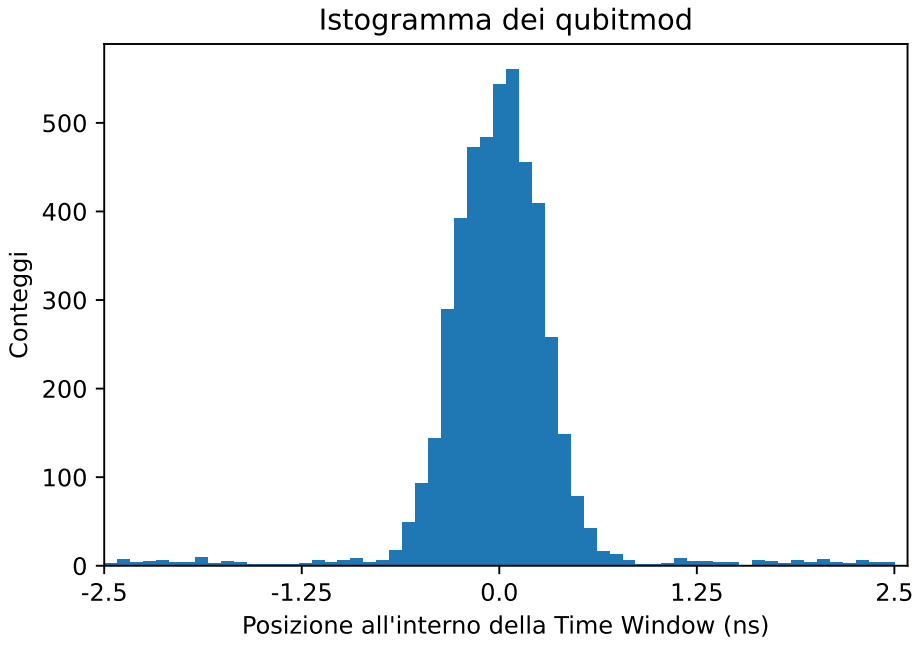


Figura 3.5: Istogramma dei qubitmod del segnale quantistico.

Aggiornamento del coefficiente di regressione lineare: Durante l'esecuzione della QKD è possibile che, a causa dei drift dei clock dei vari dispositivi coinvolti, il valore τ^b oscilli, modificando gli istanti di arrivo in cui

i qubit giungono al ricevitore. Per fare in modo che questi impulsi non vengano scartati, andando ad intaccare le performance della QKD, è necessario disporre di una stima τ_0^b il più possibile aggiornata. Per mantenere aggiornata la stima τ_0^b del periodo di arrivo degli impulsi del segnale quantistico, il protocollo Qubit4Sync prevede un meccanismo che in maniera dinamica applica dei piccoli aggiustamenti al valore di τ_0^b. Rappresentando graficamente la sequenza di arrivo dei qubitmod come in Fig.[3.6], ed applicando tecniche di regressione lineare è possibile verificare se la stima di cui si dispone attualmente di τ^b sia corretta o meno. Se la pendenza della retta ottenuta dovesse essere diversa da 0 significherebbe che la stima τ_0^b di cui disponiamo non è esatta e che per questo motivo, quando si applica la formula (3.3) si ottiene una sequenza q_n che non si distribuisce intorno ad un valore ma che si ripartisce in maniera più o meno omogenea lungo tutto il range di valori possibili. La tecnica di regressione lineare usata nel Qubit4Sync è chiamata *Least Trimmed Squares (LTS)*[20] e permette di trovare la retta che meglio si adatta all'insieme dei qubitmod. Nello specifico questo metodo si analisi dei dati mira a minimizzare la funzione obbiettivo riportata nella formula (3.4). Questa funzione è derivata dal più comune metodo *Least Squares* e quindi consiste nella somma del quadrato dei residui, limitandosi però a prendere in considerazione soltanto k degli n qubitmod di cui si dispone. In questo modo l'algoritmo, aumentando la sua complessità computazionale a causa del fatto che è necessario testare tutte le $\binom{n}{k}$ combinazioni, risulta essere più robusto all'effetto degli outliers.

$$\phi_k(\beta) = \sum_{j=1}^{k} r_{(j)}(\beta)^2 \tag{3.4}$$

Una volta ricavata la retta con pendenza m_x che minimizza la funzione obbiettivo, è possibile correggere la stima τ_0^b applicando la formula (3.5).

$$m_x = \frac{\tau^b - \tau_0^b}{\tau^b} \Longrightarrow \tau^b = \frac{\tau_0^b}{1 - m_x} \tag{3.5}$$

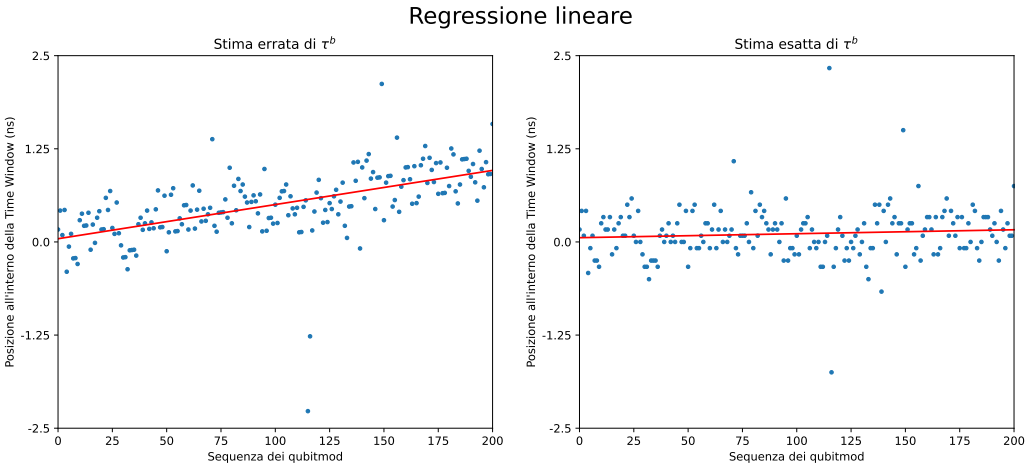


Figura 3.6: Applicazione dell'algoritmo LTS alla sequenza di timestamps del segnale.

Ripristino della fase: L'ultimo step effettuato dal protocollo Qubit4Sync è quello necessario per il recupero e la correzione della fase con cui il segnale quantistico arriva a Bob. Utilizzando ancora una volta i qubitmod ricavati negli step precedenti è possibile costruire un istogramma, esattamente come si è fatto in Fig.[3.5]. Da questo grafico osservando la posizione del massimo è possibile stimare la fase media con cui gli impulsi arrivano a Bob. Modificando la posizione della Time Window di conseguenza è possibile centrare gli impulsi all'interno dell'intervallo di campionamento minimizzando in questo modo il numero di impulsi scartati.

Il protocollo di sincronia Qubit4Sync è un'ottima soluzione che non avendo bisogno di nessuna particolare risorsa può essere applicato senza troppi problemi nella maggior parte dei problemi di sincronia. Questa sua flessibilità e semplicità di applicazione però viene pagata in termini di complessità computazionale. Il protocollo ha infatti bisogno di applicare un notevole livello di analisi ad ogni pacchetto dati che prende in considerazione ed anche se questo non va ad intaccare la qualità dei risultati che produce può rappresentare un aspetto negativo da tenere in considerazione.

3.2 Il protocollo ExtRef

Il protocollo di sincronizzazione ExtRef basa il suo funzionamento sull'utilizzo di un segnale di riferimento esterno grazie al quale ricavare valori dei parametri di sincronia stabili ed affidabili. Il supporto di un segnale chiaro e costante, anche se comporta un maggior uso di risorse ed un aumento della complessità del sistema, fornisce un grande aiuto che semplifica notevolmente l'analisi del segnale quantistico. L'approccio adottato con questo metodo prevede di considerare il più possibile inattendibile il segnale quantistico e ricavare le informazioni di periodo e fase grazie all'aiuto fornito dal segnale aggiuntivo. Nella Fig.[3.7] è riportato uno schema delle fasi principali seguite dal protocollo ExtRef.

Raccolta dei dati del segnale quantistico e di quello di riferimento: In questa prima fase del protocollo vengono raccolti i dati provenienti sia dal segnale quantistico che dal segnale di riferimento. I segnali di riferimento usati consistono in clock standard a 10MHz che, a causa della mancanza dell'elettronica necessaria alla gestione di queste frequenze da parte del TDC, vengono opportunamente sottocampionati prima di essere elaborati. Il segnale di riferimento risultante consiste in un clock a $100kHz$.

Stima del periodo: All'interno del protocollo ExtRef la stima del periodo del segnale quantistico viene effettuata basandosi sul segnale di riferimento esterno. Data l'indipendenza dei drift dei clock dei Alice e Bob, anche conoscendo le esatte frequenze a cui viene trasmesso e ricevuto il segnale quantistico queste perdono di significato, in quanto a causa dei drift che queste subiscono la stessa frequenza per Alice potrebbe non essere uguale per Bob. Conoscendo a priori la frequenza f_A con cui il segnale quantistico viene trasmesso e la frequenza f_{ext} del segnale di riferimento, è possibile però stimare il periodo di arrivo del segnale quantistico anche se non esattamente uguale a quello aspettato. Nella formula (3.6) l'unica incognita è rappresentata dalla variabile τ_{TDC} che rappresenta il periodo di campionamento del TDC. L'intera incertezza di questo calcolo viene spostata nella determinazione di questa variabile imputandola responsabile, anche se in realtà solo in parte, della variabilità del valore di τ_0^b.

$$\tau_0^b = \frac{1}{f_A\ \tau_{TDC}} \Longrightarrow \frac{1}{f_A}\ \left(\frac{\sum_{n=1}^{N-1} (t_{n+1}^{ext} - t_n^{ext})}{N\ 10^{12}\ f_{ext}^{-1}} \right) \tag{3.6}$$

Per ricavare una stima del periodo di campionamento del TDC, facendo uso del segnale di riferimento esterno, viene diviso il periodo del segnale

esterno espresso in picosecondi 10^{12} f_{ext}^{-1}, per la media di tutti gli N intervalli temporali tra un timestamp t_i^{ext} e il successivo t_{i+1}^{ext}, in questo modo si ottiene una stima del valore di τ_{TDC}. Ripetendo quest'operazione una volta per pacchetto è possibile avere un valore sempre aggiornato della stima di τ_{TDC} e di conseguenza anche di τ_0^b.

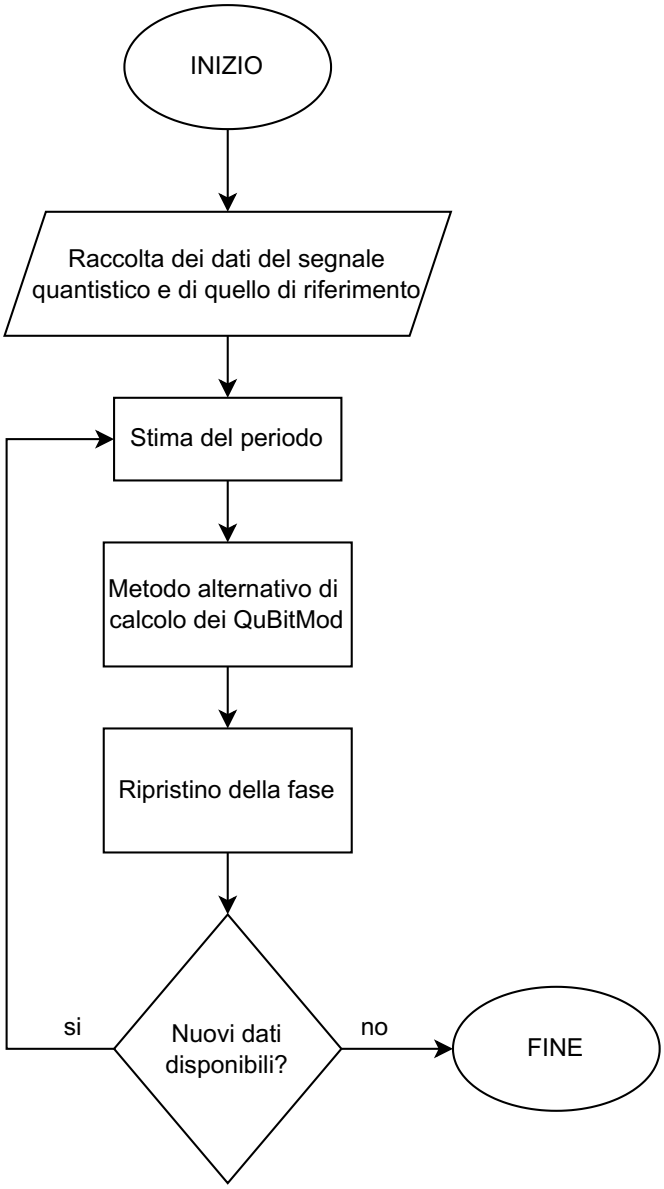


Figura 3.7: Diagramma a blocchi del funzionamento del protocollo ExtRef

Metodo alternativo di calcolo dei QuBitMod: Per ricavare la sequenza di qubitmod nel caso si faccia uso di un segnale di riferimento esterno è necessario cambiare la procedura con cui questi vengono calcolati. Nel caso del protocollo Qubit4Sync non avendo a disposizione un riferimento esterno per restare sincronizzato con gli impulsi inviati da Alice è necessario ripetere l'analisi ad ogni pacchetto ricevuto. In questo modo, data la varianza di arrivo degli impulsi quantistici e le pesanti perdite, la Time Window veniva posizionata in ogni pacchetto in maniera differente e questo comportava che fosse necessario applicare per ogni pacchetto uno shift correttivo diverso.

Disponendo di un clock esterno, un segnale molto più forte e stabile di quello quantistico, è possibile realizzare una procedura più affidabile. Centrando la Time Window usando i timestamps prodotti dal segnale di riferimento esterno, l'istogramma dei qubitmod verrà a trovarsi posizionato nello stesso modo in ogni pacchetto lo si osservi. Incaricando il segnale esterno del ruolo di riferimento temporale è possibile evitare di effettuare il controllo della fase del segnale quantistico in ogni pacchetto e limitarsi soltanto quando viene superata una certa soglia di errore, abbassando in questo modo la complessità del protocollo. Per limitare l'impatto dei drift dei clock, che anche all'interno di un singolo pacchetto possono andare ad intaccare la periodicità del segnale, si è scelto di applicare un metodo alternativo per il calcolo dei qubitmod, in modo che venga sfruttata il più possibile l'informazione portata dal segnale di riferimento. Come si osserva infatti, se l'arrivo del primo timestamp del pacchetto avviene all'istante $\Delta t_1 + \xi_1$ e l'arrivo del secondo avviene all'istante $\Delta t_2 + \xi_2$, l'arrivo del timestamps ennesimo può essere rappresentato come $\Delta t_n + \xi_n$ dove $\Delta t_n = \sum_{i=1}^{n} \Delta t_i$ ed il termine $\xi_n = \sum_{i=1}^{n} \xi_i$. Nella situazione appena presentata se il clock che regola la frequenza di attivazione del laser non dovesse essere particolarmente stabile è chiaro che la sommatoria degli errori ξ_n possa rappresentare un ostacolo ed andare ad appiattire l'istogramma dei qubitmod ed in questo modo abbassare le performance dell'intero protocollo. Per questo motivo si è scelto di prendere come riferimento temporale non un singolo timestamps del segnale esterno per l'intero pacchetto ma di calcolare i qubitmod usando di volta in volta il timestamps temporalmente precedente rispetto al impulso quantistico che si sta considerando.

$$q_n = mod_{\tau_0^b}(t_n - t_j^{ext}) \qquad n = 1, \ldots, N \qquad t_j^{ext} : \quad t_j^{ext} < t_n < t_{j+1}^{ext} \tag{3.7}$$

Aggiornare il timestamp di riferimento man mano vengono considerati tutti gli impulsi quantistici contenuti in un pacchetto è un piccolo accorgimento che ci permette di troncare la sommatoria degli errori $\sum_{i=1}^{n} \xi_i$ evitando

che la statistica degli arrivi venga intaccata eccessivamente dalle variazioni del clock.

Ripristino della fase: Allo stesso modo visto per il protocollo Qubit4Sync per recuperare l'informazione sulla fase degli impulsi quantistici viene fatto uso dell'istogramma prodotto dai Qubitmod. Questa volta però è stato scelto di utilizzare il valore medio invece del valore massimo. Quest'ultimo infatti essendo soggetto ad oscillazioni statistiche non si rivela la scelta migliore per un protocollo che fa della stabilità il suo obbiettivo primario.

Il protocollo ExtRef utilizzando un segnale aggiuntivo riesce nel suo obbiettivo di abbassare il livello di analisi del segnale per stabilire la sincronia tra Alice e Bob. I risultati prodotti però sono buoni soltanto in circostanze limitate. Per funzionare infatti questo protocollo necessita che ad entrambi i dispositivi venga fornito il medesimo segnale di riferimento temporale su cui basare l'invio e la ricezione degli impulsi ottici. Ed anche in queste condizioni siccome il protocollo applica soltanto una correzione periodica necessita che il segnale di riferimento conservi un alto grado di stabilità per funzionare al meglio. E' evidente che malgrado l'obbiettivo di riduzione della complessità computazionale sia stato raggiunto le possibilità applicative di questo protocollo siano limitate e che l'applicabilità sia realizzabile solo negli scenari dove queste risorse siano disponibili.

3.2.1 Clock fornito dal sistema GNSS

Il protocollo riportato nelle sezioni precedenti è stato sviluppato in laboratorio e prevede che sia possibile, aggiungendo un collegamento fisico, fornire ad Alice e Bob uno stesso segnale di riferimento su cui basare il proprio funzionamento. Questo scenario però non è molto comune, in laboratorio si tratta di stendere pochi metri aggiuntivi di cavo ma fuori da quel contesto disporre di un collegamento ulteriore e non strettamente necessario può non essere possibile o quantomeno essere molto costoso. Per questo motivo per facilitare l'applicabilità di questo protocollo è stata realizzata una variante che fa uso del segnale di clock fornito dal sistema GNSS (Global Navigation Satellite System).

La sigla GNSS, si riferisce ai tre sistemi satellitari GPS, GLONASS e Galileo, che possono essere utilizzati congiuntamente, ottenendo significativi benefici, in particolare nelle zone in cui le operazioni sono limitate da ostacoli naturali o artificiali. Queste reti di satelliti sono in grado di offrire molteplici servizi, alcuni dei più utilizzati sono: posizionamento di mezzi in navigazione,

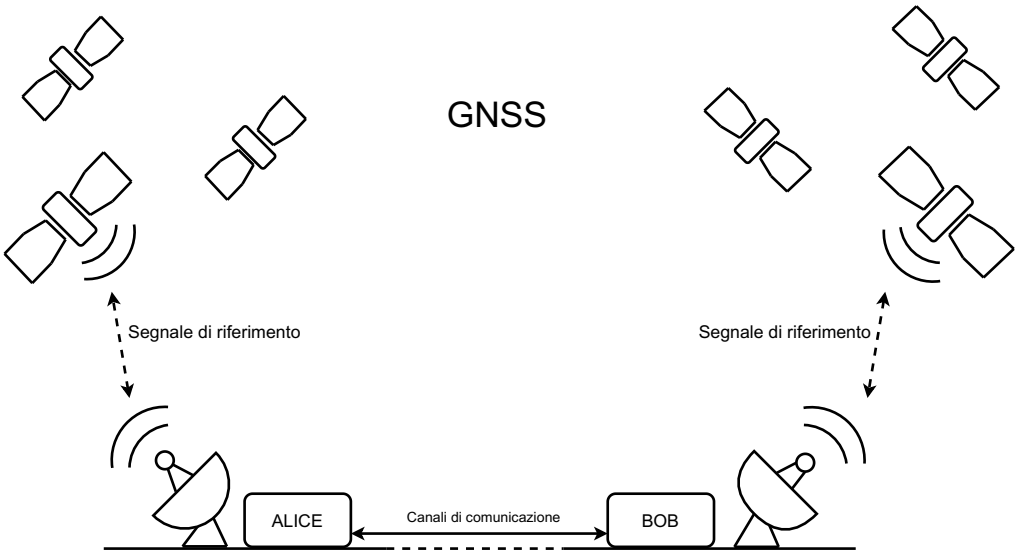


Figura 3.8: Scenario di applicazione del protocollo ExtRef che fa uso del segnale satellitare

in cielo, terra ed acqua, rilievi topografici, rilievi cinematici di alta precisione, fotogrammetria, monitoraggio delle deformazioni crostali e molti altri.

Il funzionamento di base di questi sistemi di navigazione, si basa su una rete di decine di satelliti orbitanti intorno alla Terra ad un'altezza di circa 20.000 km. Per poter definire la posizione di un ricevitore sulla superficie terrestre, questi satelliti devono essere sincronizzati con il ricevitore ed in seguito emettere uno specifico segnale. A questo punto per il ricevitore è possibile registrare gli intervalli temporali che intercorrono tra l'emissione del segnale da parte dei satelliti e l'arrivo dei vari segnali provenienti da diversi satelliti. In questo modo, conoscendo a priori la precisa orbita di ciascun satellite, se si dispone di un numero di satelliti sufficiente si è in grado di determinare con precisione la posizione nello spazio del ricevitore.

Per poter portare a termine con successo questa procedura ci sono molteplici fonti di errore ed effetti relativistici che devono essere considerati, per questo motivo è richiesto un altissimo livello di precisione nelle misure temporali. I satelliti che fanno parte della costellazione del sistema GNSS sono attrezzati con orologi atomici in grado di garantire un'accuratezza migliore di un miliardesimo di secondo al giorno.

Sfruttando la disponibilità del sistema di orologi sincronizzati offerto dal GNSS è possibile sviluppare un'implementazione del protocollo ExtRef e fare in modo che Alice e Bob ricevano lo stesso riferimento riuscendo a temporizzare l'invio e la ricezione degli impulsi luminosi anche senza aggiungere

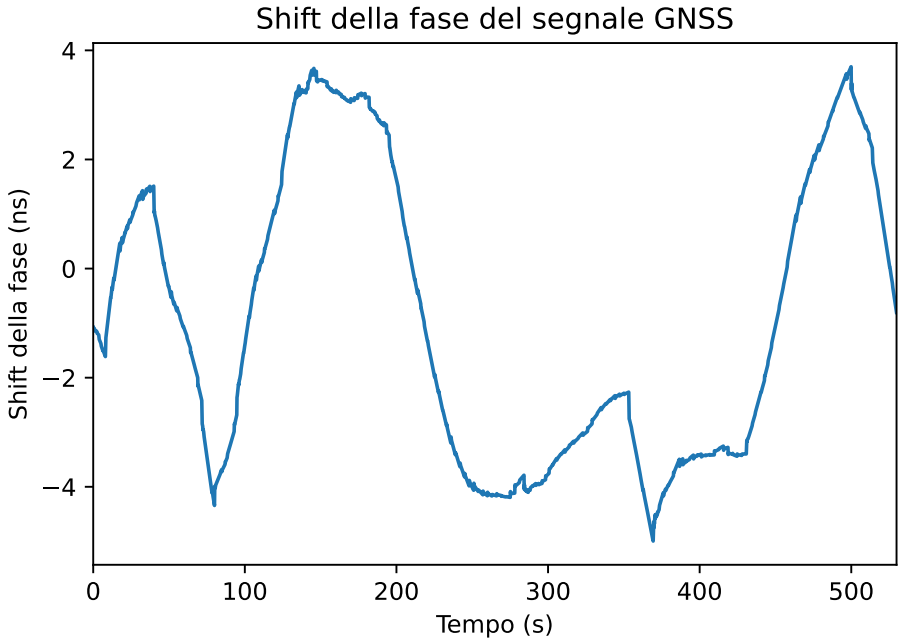


Figura 3.9: Rappresentazione del cambiamento della fase dei clock di Alice e Bob quando è usato il segnale del GNSS

ulteriore cablaggio.

Dotando Alice e Bob di un ricevitore GNSS in grado di sincronizzare il proprio clock con quello fornito dal sistema, esattamente come illustrato in Fig[3.8], è possibile ottenere un segnale di riferimento estremamente preciso. I ricevitori GNSS sono dispositivi dotati di un'antenna in grado di ricevere il segnale di riferimento emesso dai satelliti ed allinearsi ad esso. Essendo apparecchiature molto più economiche, i ricevitori GNSS non sono in grado di garantire la stessa accuratezza degli orologi atomici presenti sui satelliti, e per questo anche dopo essersi allineati al segnale emesso dai satelliti periodicamente vengono applicati dei controlli e delle correzioni per mantenere il clock in uscita dai ricevitori il più simile possibile rispetto a quello emesso dal sistema GNSS. Questi piccoli errori nell'allineamento dei due segnali, si traducono in cambiamenti della fase del segnale di riferimento che possono arrivare a valori anche di qualche nanosecondo, come è possibile osservare in Fig.[3.9]. Il fatto che le variazioni della fase siano così ampie e che possano estendersi così a lungo nel tempo non era un'eventualità prevista quando è stato sviluppato il protocollo ExtRef. In queste condizioni una correzione puntuale della fase attuata periodicamente non è sufficiente per mantenere una buona sincronia. Per questo motivo è stato necessario applicare delle

aggiunte al protocollo ExtRef per fare in modo che possa rivelarsi adeguato anche in questo contesto.

Modello Predittivo: A causa delle imperfezioni nell'allineamento tra il clock fornito dai satelliti e quelli in output dai ricevitori GNSS, tra i segnali di riferimento di Alice e Bob vengono a formarsi delle variazioni nella fase la cui ampiezza e durata sono notevoli. Le correzioni puntuali applicate dal protocollo ExtRef non sono sufficienti a mantenere la sincronia tra Alice e Bob. Per raggiungere quest'obbiettivo è stato necessario implementare all'interno del protocollo ExtRef un modello predittivo in grado di anticipare il valore del cambiamento della fase anche negli istanti dove questa non viene misurata direttamente. Lo sviluppo di questa tecnica ha consentito di ottenere un aggiustamento continuo dello shift della fase del segnale, riuscendo a mantenere lo spostamento degli impulsi luminosi limitato all'interno dell'intervallo di campionamento.

Nella pratica è stato aggiunto all'interno della fase di *Phase Recovery* un record che tenga traccia delle ultime osservazioni fatte sulla fase del segnale quantistico, e utilizzando queste informazioni viene effettuato un fit lineare che permette di prevedere, con buona precisione, quali saranno i prossimi valori dello spostamento della fase anche senza osservarli direttamente.

La decisione di adottare un fit lineare basato su un numero limitato di punti e di non spingersi verso soluzioni più elaborate, come ad esempio i modelli polinomiali, è stata presa considerando il fatto che le oscillazioni nella fase dei segnali di riferimento sono in genere molto lunghe, se analizzate sulla scala dei singoli pacchetti, e per questo motivo soluzioni più complesse si sarebbero rivelate più sensibili ad effetti rumorosi non portando nessun effettivo beneficio al modello.

In Fig.[3.10] viene riportato il confronto tra la tecnica usata precedentemente dal protocollo ExtRef e quella ideata per l'utilizzo del segnale fornito dal GNSS. Data l'aggiunta del modello predittivo questa nuova versione del protocollo è stata nominata *Predictive ExtRef*. Osservando il comportamento del protocollo ExtRef nel grafico, si nota che dopo la correzione della fase in un pacchetto quelli successivi subiscono un progressiva degradazione della sincronia che causa gli impulsi ricevuti da Bob a muoversi all'interno della Time Window comportando una possibile perdita di qubit. Come si può osservare invece il Predictive ExtRef avendo sempre a disposizione un valore aggiornato, per stima o per misurazione, della fase del segnale quantistico riesce a restare più aderente alla reale variazione della fase. In questo modo anche nei pacchetti che si trovano fra un'effettiva misurazione della fase e la successiva gli impulsi ottici rimangono centrati.

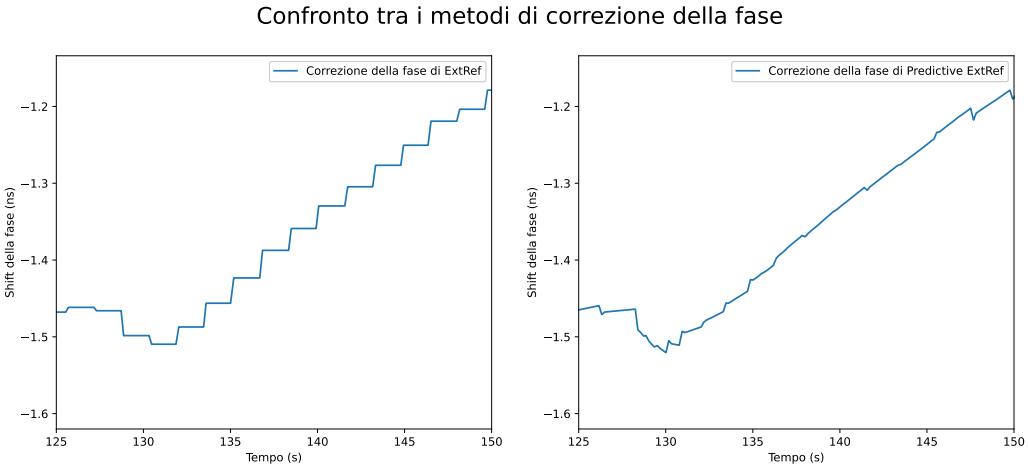


Figura 3.10: Confronto tra i metodi usati dal protocollo ExtRef e dal Predictive ExtRef per correggere la fase del segnale quantistico

Ottimizzazione della frequenza di controllo della fase: Una conseguenza del fatto che la fase del segnale di riferimento vari tra valori così diversi tra loro in maniera così lenta, è che la strategia di controllo periodico della fase si rivela inefficiente. Come è facile dedurre osservando la Fig.[3.9], ci sono momenti in cui è necessario che il controllo della fase avvenga in maniera frequente, ed altri invece dove grazie al modello predittivo non è necessario mantenere la stessa frequenza. Allo scopo di ottimizzare il numero di controlli da effettuare sulla fase del segnale quantistico è stato implementato un algoritmo di *Additive Increase/Multiplicative Decrease (AIMD)*. Il più famoso algoritmo di questa famiglia è quello utilizzato per gestire l'ampiezza della congestion window nel protocollo TCP/IP[21]. In quel contesto lo scopo dell'algoritmo è quello di sondare la capacità della rete Internet per capire qual'è il numero massimo di pacchetti IP che può essere inviato senza il rischio di congestionare il sistema. Applicato nel nostro scenario invece questo algoritmo si rivela utile per gestire la frequenza di controllo della fase del segnale quantistico. Nei momenti dove lo shift del segnale quantistico è costante il controllo della fase può essere effettuato sporadicamente. Aumentando l'ampiezza dell'intervallo di controllo della fase, è possibile ridurre il livello di analisi del segnale abbassando la complessità computazionale dell'intero protocollo. Grazie all'implementazione di questa tecnica inoltre è possibile intensificare il controllo del segnale quantistico in quei momenti dove la fase sta cambiando in maniera inaspettata.

L'algoritmo si compone di tre passaggi: aumento additivo, decremento moltiplicativo e slow-start.

- **Aumento Additivo:** In questa prima fase l'algoritmo sonda la stabilità del segnale quantistico, aumentando lentamente l'intervallo temporale tra un controllo e il successivo. In questo modo il controllo della fase del segnale viene effettuato su sempre meno pacchetti cercando di portare questo numero al suo minimo.

- **Decremento Moltiplicativo:** Se durante la fase di Aumento Additivo viene rilevato che si sta peggiorando la sincronia tra Alice e Bob, la prima fase viene prontamente interrotta, il valore raggiunto dell'ampiezza dell'intervallo viene memorizzato e l'algoritmo ripristina l'ampiezza dell'intervallo di controllo al valore iniziale.

- **Slow-Start:** In questa fase l'algoritmo cerca di ritornare rapidamente a valori dell'ampiezza dell'intervallo di controllo elevati. Usando l'informazione memorizzata nella fase di Decremento Moltiplicativo viene impostata una soglia di sicurezza, chiamata *Slow-Start Threshold*, in genere uguale ad una frazione del valore memorizzato. Da questo momento l'algoritmo procederà con velocità esponenziale fino al raggiungimento di quella soglia e da lì in poi ripeterà la fase di Aumento Additivo.

In Fig.[3.11] è riportata una misurazione dell'ampiezza del intervallo di controllo della fase durante un'esecuzione del protocollo Predictive ExtRef. Come è possibile osservare si sono raggiunti intervalli di controllo di ampiezza 3 secondi riducendo notevolmente il carico computazionale necessario a mantenere la sincronia tra Alice e Bob.

Il protocollo Predictive ExtRef grazie alle sue aggiunte riesce a sopperire alla mancanze presenti nel protocollo ExtRef. Con l'implementazione del modello predittivo e del algoritmo di gestione degli intervalli di controllo, riesce nell'intento di minimizzare il livello di analisi del segnale richiesto continuando a produrre risultati di qualità. Essendo sviluppato per lavorare con il segnale del GNSS il protocollo ha acquisito robustezza riuscendo a gestire segnali di riferimento reali che possono a loro volta essere soggetti ad oscillazioni della frequenza. Grazie a queste sue caratteristiche lo spettro di applicazione di questo protocollo si è allargato notevolmente diventando un protocollo di interesse.

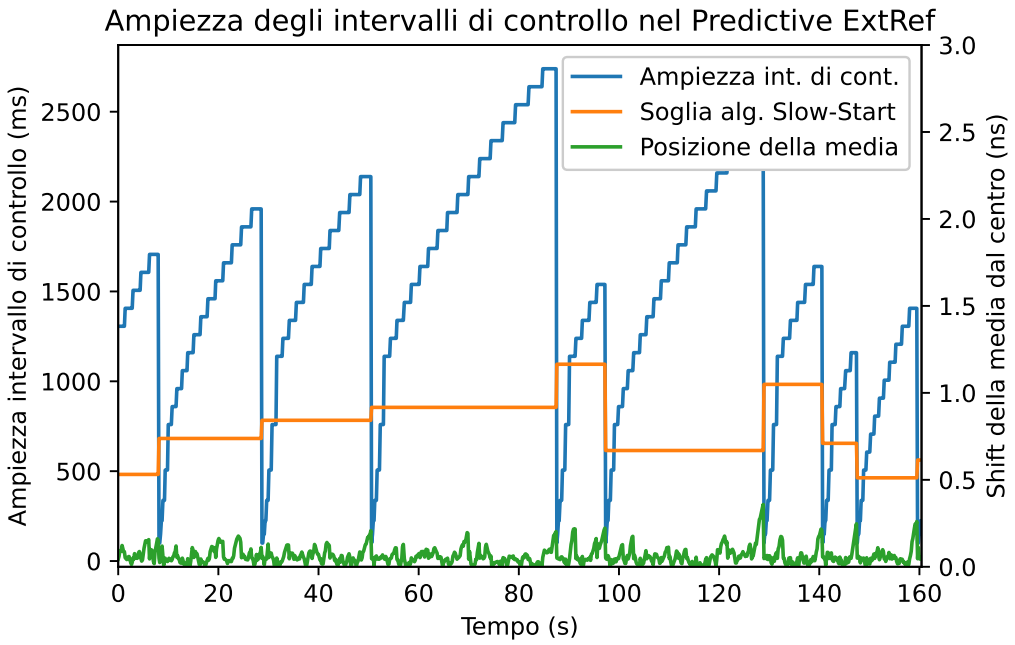


Figura 3.11: Rappresentazione del algoritmo applicato dal Predictive ExtRef per gestire l'ampiezza del intervallo di controllo della fase del segnale.

3.3 Metodi di allineamento

Una volta recuperate le informazioni sul periodo e sulla fase del segnale quantistico siamo riusciti ad ottenere due sistemi che si muovono in maniera analoga l'uno rispetto all'altro. Tra i due però potrebbe insistere una mancata corrispondenza nell'indicizzazione degli impulsi ottici. Chiariamo questo punto con un esempio: data la sensibilità del segnale quantistico, come abbiamo detto nelle sezioni precedenti, soltanto un fotone ogni mille circa riesce a raggiungere e ad essere rilevato da Bob, per questo motivo quando il ricevitore inizia a numerare gli impulsi che gli sono arrivati, questa numerazione potrebbe non coincidere con quella utilizzata da Alice per l'invio degli impulsi. Se Bob e Alice non condividono l'indicizzazione dei qubit sarà per loro impossibile portare a termine la procedura di sifting e quindi produrre la raw key. Per ovviare a questo problema, prima della trasmissione degli stati quantistici usati per produrre la chiave, viene portata a termine una procedura di allineamento che grazie alle tecniche che verranno presentate nelle prossime sezioni, permettono a Bob di ricavarsi la stessa indicizzazione degli impulsi usata da Alice.

3.3.1 Stringa di sincronia nota

Uno degli strumenti più utilizzati in questo contesto è la stringa di sincronia. Dati i requisiti presenti, ossia un livello di precisione dell'ordine del nanosecondo, soluzioni più tradizionali e più largamente utilizzate come il *Network Time Protocol (NTP)*[22] e il *Precision Time Protocol (PTP)*[23], non sono applicabili in quanto non in grado di garantire una precisione adeguata. Una stringa di sincronia consiste in una sequenza di bit nota a priori sia ad Alice che a Bob. Questa sequenza, tipicamente molto lunga (milioni di bit), viene inviata all'inizio della trasmissione da Alice e una volta che Bob l'ha ricevuta viene applicata al segnale la cross-correlazione con il file originale riuscendo anche in presenza di una notevole quantità di errore a determinare l'offset presente tra gli indici di Alice e Bob.

$$R_{xy}[n] = (x \star y)[n] \overset{\text{def}}{=} \sum_{m=-\infty}^{\infty} x^*[m]\; y[n+m] \tag{3.8}$$

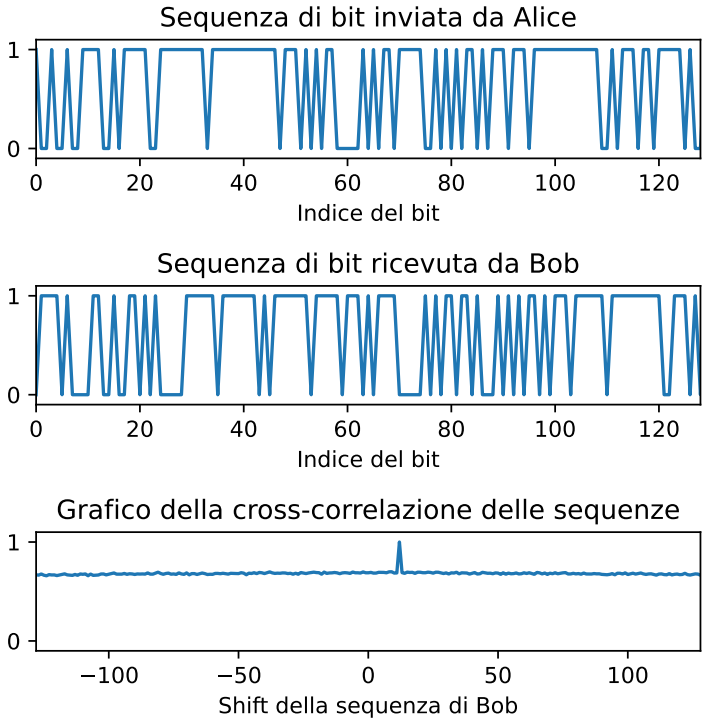


Figura 3.12: Rappresentazione grafica della stringa di sincronia inviata da Alice, della sequenza di bit ricevuta da Bob e della loro cross-correlazione.

In Fig.[3.12] si può osservare che la sequenza di bit trasmessa da Alice è leggermente diversa da quella ricevuta da Bob, e soprattutto non sia allineata. Malgrado ciò, dato che questa sequenza contiene la stringa di sincronia, grazie all'applicazione della cross-correlazione da parte di Bob è possibile ricavare lo shift da applicare alla numerazione adottata da quest'ultimo per fare in modo che coincida con quella che sta usando Alice.

3.3.2 Stringa di sincronia casuale condivisa durante l'esecuzione

Per permettere l'applicazione di protocolli entanglement-based si è sviluppata una variante del metodo di allineamento che fa uso della stringa di sincronia. In uno scenario di applicazione della QKD dove gli stati quantistici sono forniti ad Alice e Bob da una sorgente esterna al sistema, non è possibile imporre la trasmissione di una sequenza nota. Per questo motivo si è prevista la possibilità che sia Alice, dopo aver ricevuto gli stati trasmessi dalla sorgente a condividerli con Bob, dando in questo modo la possibilità agli utenti in ascolto di allineare le loro sequenze di stati quantistici. Nella pratica è stata resa disponibile una configurazione per Bob nella quale durante l'esecuzione del protocollo resta in attesa sul canale autenticato che gli venga condivisa la sequenza di stati da utilizzare come stringa di sincronia. Una volta ricevuta applicando la sequenza agli stati quantistici ricevuti dalla sorgente, anche in presenza di una significativa porzione di mismatch, Alice e Bob dovrebbero essere in grado di allineare i propri indici di ricezione degli impulsi quantistici.

3.3.3 Pulse per second (PPS)

Un altro strumento molto utile per allineare le sequenza degli impulsi di Alice e Bob è il segnale *Pulse Per Second (PPS)*. Il segnale PPS produce un impulso ogni secondo e può essere fornito da diverse fonti, come orologi atomici e sistemi GNSS. Programmando Alice in modo che inizi la comunicazione degli stati quantistici in sincronia con il segnale PPS è possibile diminuire di molto lo sforzo necessario per allineare le sequenze degli impulsi di Alice e Bob. Grazie all'utilizzo di questo segnale, anche se non si può eliminare l'uso della stringa di sincronia, è invece possibile ridurre considerevolmente la sua lunghezza, abbassando notevolmente il costo computazionale dell'applicazione della cross-correlazione. A causa di scelte di progettazione si è scelto di non investigare le possibilità offerte da questo metodo di sincronizzazione perché già testate dal gruppo QuantumFuture. Nella mia esperienza nei laboratori mi sono concentrato nello sviluppo di protocolli di sincronia alternativi e

nella realizzazione di una configurazione aggiuntiva per la realizzazione di scenari di QKD con sorgente di fotoni entangled.

Capitolo 4

Analisi dei risultati

In questo capitolo viene presa in considerazione un'ampia gamma di metriche per valutare le performance dei vari protocolli di sincronia. Lo scopo di quest'analisi è quello di evidenziare eventuali differenze, risaltando quale genere di protocollo si rivela essere il più adatto e in quali situazioni.

Tutti i test riportati sono stati effettuati cercando di mantenere il più possibile simili le condizioni e i valori dei parametri iniziali, in modo da ottenere dei risultati comparabili.
Per valutare le diverse strategie di sincronia sono stati messi a confronto i risultati prodotti dai vari protocolli, le principali grandezze prese in esame sono i qubitmod e le quantità di segnale e rumore riconosciute dai vari algoritmi. In Fig.[4.1] è riportato un esempio dei grafici utilizzati per analizzare il comportamento dei qubitmod, in modo da far prendere confidenza al lettore con le rappresentazioni dei risultati e le loro interpretazioni. Come si avrà modo di osservare, analizzare i qubitmod fornisce informazioni sull'efficacia della tecnica applicata sia sul breve che sul lungo periodo.

Il grafico della sequenza temporale dei qubitmod è utile per riconoscere la qualità del segnale quantistico in ingresso e per capire se la stima del periodo di quest'ultimo è stata effettuata correttamente. Come si vede dal grafico riportato come esempio, il canale quantistico preso in considerazione non appare essere affetto da nessun particolare disturbo e di conseguenza il segnale riesce a raggiungere il ricevitore evitando di manifestare evidenti buchi, è inoltre presente una componente rumorosa riconoscibile dagli sporadici picchi, sia positivi che negativi, presenti nella sequenza di qubitmod. La stima del periodo appare tuttavia corretta in quanto l'intera sequenza sembra oscillare attorno allo zero senza manifestare un'evidente tendenza ad allontanarsi da esso.

Il secondo grafico riportato invece, malgrado sia costruito con gli stessi dati, mette in evidenza proprietà differenti. Il lag plot viene costruito ana-

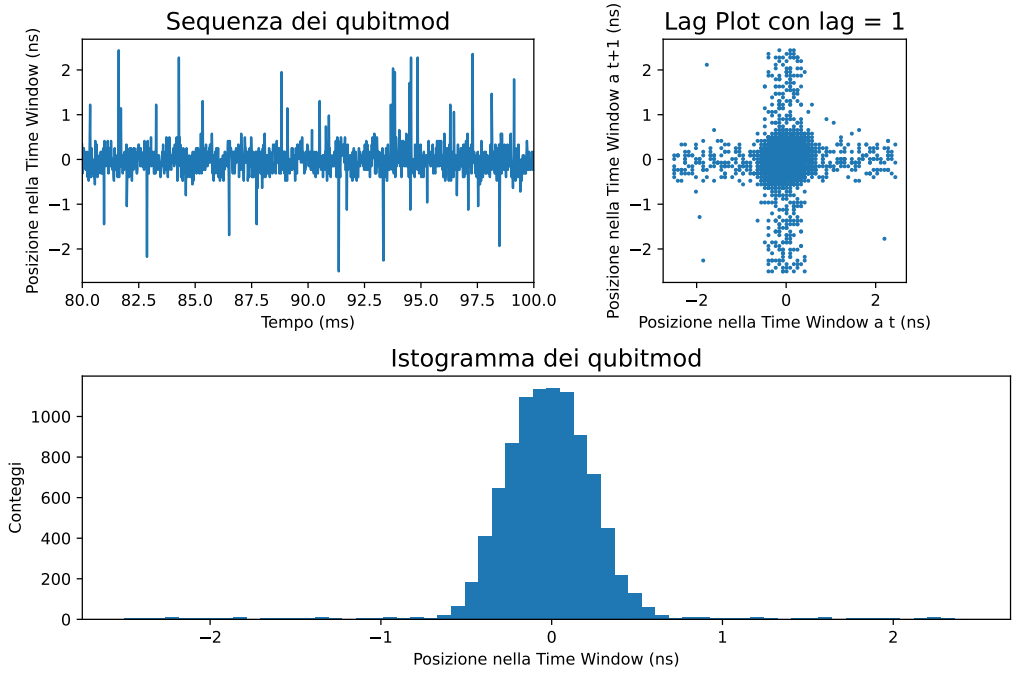


Figura 4.1: Composizione di grafici rappresentanti le caratteristiche statistiche dei qubitmod

lizzando di volta in volta non un singolo qubitmod ma una coppia tale che la distanza nella sequenza dei qubitmod tra questi due dati sia uguale al lag. In questo caso siccome stiamo osservando il lag plot di primo ordine, nel grafico viene disegnato un punto di coordinate (x_t, x_{t+1}) per ogni coppia di qubitmod tale che uno è il successivo dell'altro all'interno della sequenza.

Questo genere di grafico mette in evidenza le dipendenze dei dati, rivelando l'esistenza di eventuali correlazioni e oscillazioni periodiche. Come si può osservare nell'esempio i qubitmod del segnale sono per la maggior parte posizionati al centro della finestra temporale formando una nuvola di punti posizionata al centro del grafico. La presenza delle due fasce, quella verticale e quella orizzontale, invece evidenziano il fatto che in genere se un qubitmod si trova a non essere centrato nella Time Window il successivo in invece lo è. Grazie a quest'osservazione si può concludere che generalmente l'effetto del rumore sul canale quantistico si manifesta causando sporadici picchi nella sequenza dei qubitmod. Infine, come si può vedere dal grafico, sono presenti anche alcuni punti che non risiedono né nella zona centrale né lungo le due fasce, ma che occupano lo spazio interno ai quadranti del grafico, questi punti indicano la presenza di due qubitmod successivi che si trovano decentrati rispetto alla Time Window. La loro esistenza estremamente limitata

denota il fatto che il rumore di fondo sul canale quantistico sebbene presente non costituisca un reale ostacolo ai protocolli di sincronizzazione, in quanto non è in grado di alterare significativamente l'andamento della sequenza dei qubitmod.

Per ultimo, l'istogramma dei qubitmod, già introdotto precedentemente in questa tesi, è utile per studiare il comportamento a lungo termine e caratterizzare la statistica dei qubitmod. Come si osserva nel grafico, gli impulsi ottici non arrivano a Bob posizionandosi tutti alla stessa maniera all'interno della Time Window, come discusso precedentemente a causa dei vari effetti dovuti ai drift dei clock, questi si distribuiscono intorno al centro dell'intervallo seguendo una distribuzione gaussiana. Studiando le caratteristiche di questa distribuzione è possibile comprendere se il protocollo di sincronia esaminato stia facendo o meno il suo lavoro. Idealmente vorremmo che tutti gli impulsi arrivassero nella medesima posizione, il centro dell'intervallo temporale, in questo modo avremmo la possibilità di filtrare efficacemente il rumore conservando allo stesso tempo l'intero segnale quantistico. Analizzando quindi la media e la varianza dell'istogramma è possibile confrontare le performance dei vari protocolli di sincronia.

Avendo introdotto gli strumenti tramite i quali verrà condotta la analisi dei risultati, nelle prossime sezioni verranno confrontate le performance dei protocolli di sincronia presentati in questa tesi.

4.1 Analisi delle principali misure statistiche

Il primo grafico presentato in questa analisi mette a confronto i livelli di segnale e rumore che sono stati ottenuti durante l'esecuzione dei vari protocolli di sincronia presentati in questa tesi. Prima di procedere però è necessario fare una precisazione, insieme al protocollo Qubit4Sync e alle due versioni del protocollo ExtRef è stato preso in considerazione anche lo stesso protocollo Qubit4Sync applicato nello stesso scenario del protocollo Predictive ExtRef, ossia quello dove i clock di Alice e Bob sono forniti dal sistema GNSS. Questo è stato fatto per cercare di avere un confronto il più completo e corretto possibile tra i protocolli che così facendo possono essere divisi in due categorie. La prima dove vengono utilizzati segnali soggetti ad un drift minimo e la seconda dove a causa delle problematiche discusse precedentemente gli shift dei segnali possono raggiungere valori di qualche nanosecondo.

Come si può osservare in Fig.[4.2], malgrado i livello di segnale rilevato sia lo stesso per ogni protocollo, sono presenti delle evidenti differenze. Il protocollo Qubit4Sync riesce a rilevare un ottimo valore del segnale quantistico, ma a causa del fatto che per restare sincronizzato deve ripetere l'analisi

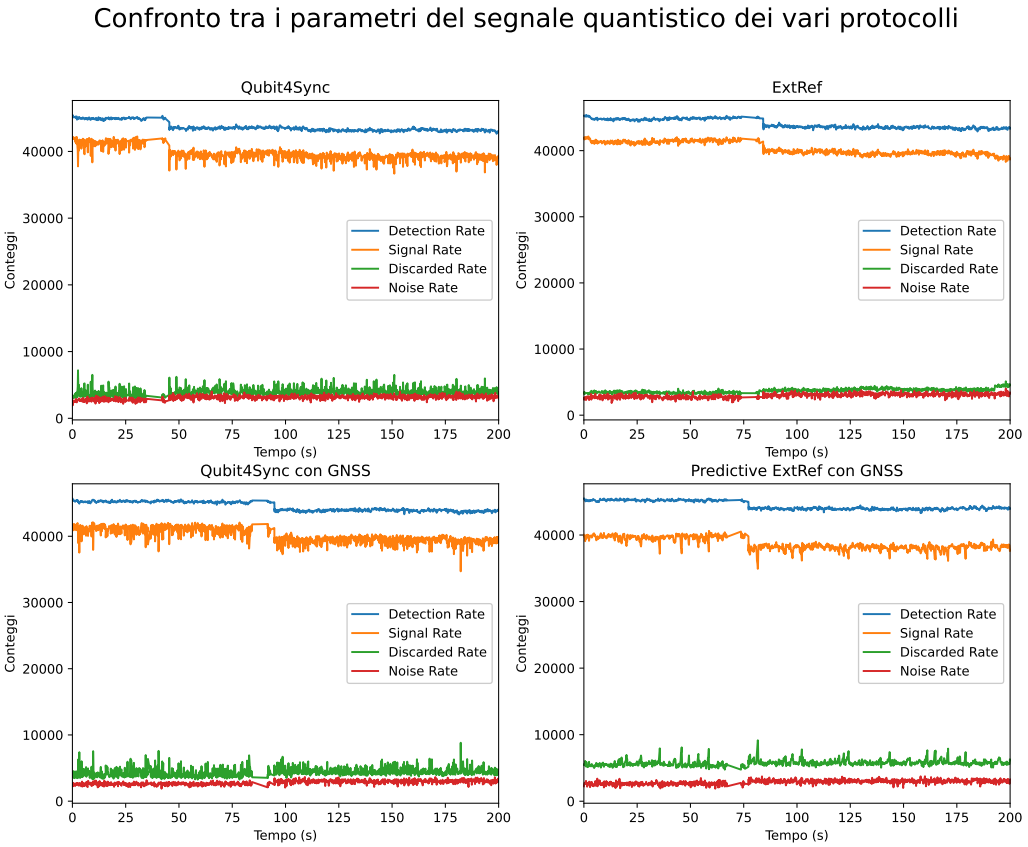


Figura 4.2: Confronto tra le metriche riguardanti il segnale quantistico rilevate dai vari protocolli.

sul segnale per ogni pacchetto, sono presenti visibili oscillazioni, sia a livello di segnale rilevato che di segnale scartato, che anche se non compromettono l'efficacia di questo protocollo intaccano la sua stabilità.

Al contrario invece, i valori del protocollo ExtRef sono molto più stabili, anche se l'analisi sul segnale è ripetuta molto più di rado. La stabilità dei suoi parametri è dovuta al fatto che Bob riceve direttamente una copia del clock di riferimento di Alice, in questo modo gli errori dovuti ad eventuali drift vengono compensati propagando il segnale anche a Bob. Come abbiamo già ripetuto varie volte, questo tra tutti i protocolli presentati è quello più difficile da realizzare in quanto richiede risorse costose e non sempre disponibili. Da notare inoltre come questo protocollo tra tutti riesca ad ottenere i valori più bassi di segnale scartato.

La seconda coppia di grafici invece è stata ottenuta introducendo il segnale di riferimento fornito dal sistema GNSS, come è possibile osservare aver dato

questo segnale ad Alice e Bob non ha modificato in maniera evidente le performance del Qubit4Sync che così facendo si conferma essere un protocollo flessibile e robusto che è possibile applicare in una molteplicità di contesti.

Viceversa per poter ottenere dei buoni risultati e seguire la filosofia del protocollo ExtRef è stato necessario fare delle aggiunte ed introdurre il protocollo Predictive ExtRef. Questo protocollo, come si osserva dal grafico, riesce ad ottenere dei buoni risultati. Le performance ottenute sono molto vicine a quelle del protocollo ExtRef per stabilità, è evidente però che sporadicamente siano presenti delle perdite, queste sono dovute al continuo ed imprevedibile cambiamento di fase tra i ricevitori GNSS di Alice e Bob. Da notare inoltre che il livello di segnale scartato è sensibilmente superiore a quelli osservati negli altri protocolli, questo è dovuto sempre alle difficoltà di allineamento tra i clock dei due dispositivi.

Infine, tutti e quattro i grafici presentano uno scalino sull'andamento del segnale ricevuto, questo segna il passaggio del protocollo dalla fase di allineamento a quella della generazione della chiave grezza e non è dovuto ai meccanismi interni degli algoritmi di sincronizzazione.

In Fig.[4.3] sono riportati i boxplot dei vari protocolli rappresentati la statistica di alcuni pacchetti dati del segnale quantistico ricevuto. Questo tipo di grafici è molto usato in statistica perché permette di avere una visione d'insieme della distribuzione seguita dai dati, mettendo in risalto alcune delle grandezze di maggiore importanza come: la media, la mediana e i quartili. Da questo grafico si può osservare come più della metà degli impulsi luminosi di ogni protocollo si trovino all'interno del intervallo di campionamento lasciando al di fuori una porzione del primo e del quarto quartile. La media e la mediana di ciascun protocollo si trovano ad essere allineate con il centro della Time Window mostrando come effettivamente tutti i protocolli riescano nell'intento di sincronizzare gli impulsi. Sono inoltre presenti alcune differenze che rispecchiano le osservazioni fatte sul precedente grafico delle metriche del segnale quantistico. Si può notare infatti che la distribuzione del protocollo ExtRef sia la più stretta e ricada quasi interamente all'interno del intervallo di campionamento, mentre quella del protocollo Predictive ExtRef si trovi ad essere la più larga, con i baffi del boxplot che escono notevolmente dall'intervallo di campionamento causando lo scarto di una porzione del segnale quantistico.

A causa dei drift dei clock dei vari dispositivi, analizzando pacchetti differenti il grafico dei boxplot potrebbe cambiare, per questo motivo per avere una rappresentazione più accurata del comportamento dei diversi protocolli è stato studiato anche l'andamento a lungo termine.

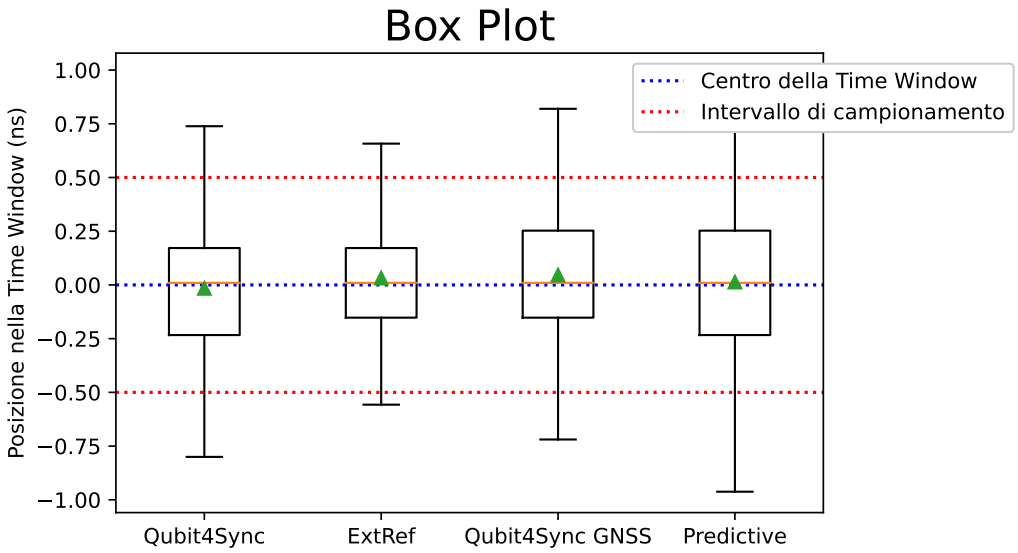


Figura 4.3: Rappresentazione dei dati in uscita ai vari protocolli di sincronia.

In Fig.[4.4], sono stati riportati gli istogrammi di ciascun algoritmo rappresentanti le posizioni degli impulsi quantistici una volta terminata la fase di sincronizzazione. La mole dei dati presi in considerazione per la realizzazione di questo grafico copre un'intera esecuzione di durata approssimativa 160 secondi. Aver scelto di raccogliere ed analizzare questa quantità di dati ci permette di valutare il comportamento a lungo termine dei vari protocolli di sincronia. Al contrario del grafico dei boxplot infatti, dove era difficile dare un giudizio a causa della variabilità dei risultati da pacchetto a pacchetto, questa rappresentazione ci permette di fare delle osservazioni interessanti.

Se considerata la prima coppia di protocolli, ossia il Qubit4Sync e l'ExtRef, non sono visibili differenze a livello di variazione delle posizioni occupate dagli impulsi ottici. Entrambi i protocolli riescono a conservare un ottimo livello di segnale quantistico mantenendo al contempo sincronizzati i dispositivi. Guardando il valore della media dell'istogramma di questi due primi protocolli si nota una differenza, anche se di un valore minimo il protocollo Qubit4Sync è più lontano dal valore centrale della Time Window rispetto alla controparte ExtRef. Questo fatto può essere spiegato ricordando che il protocollo Qubit4Sync utilizza il valore del massimo dell'istogramma per allineare i clock di Alice e Bob, questo valore oltre ad essere soggetto ad oscillazioni statistiche è inoltre limitato dal fatto che l'istogramma è formato da un numero di bin fisso, questo comporta che i valori dello shift correttivo da applicare siano limitati e difficilmente coincideranno con il valore esatto.

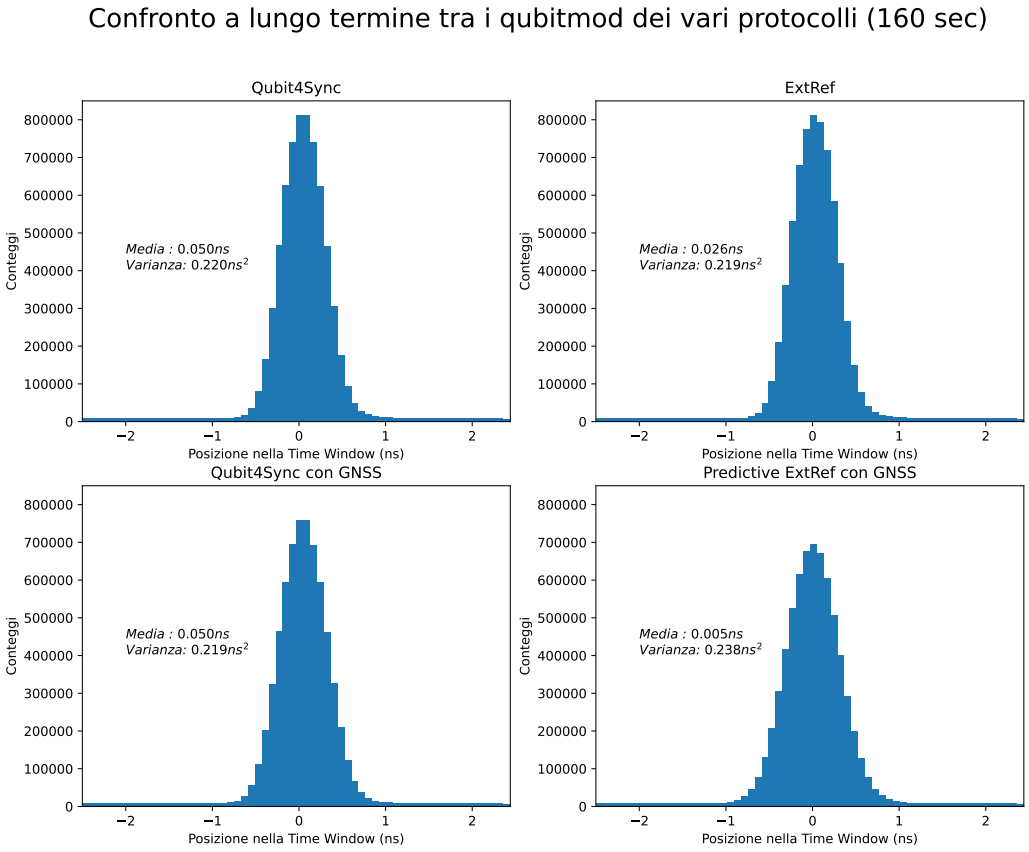


Figura 4.4: Confronto a lungo termine delle performance dei vari protocolli.

Considerando la seconda coppia di grafici, quella relativa al segnale di riferimento fornito dal GNSS, le differenze sono più evidenti. Come era facile anticipare infatti la distribuzione ottenuta dal protocollo Predictive ExtRef risulta essere meno concentrata attorno al valore centrale se paragonata a quella del Qubit4Sync. La causa di questo fenomeno è da imputare, come già detto, al fatto che lo shift relativo tra i segnali di riferimento forniti ad Alice e Bob dal sistema GNSS può arrivare a valori di parecchi nanosecondi, ed essendo imprevedibili anche se è stato implementato un modello predittivo le conseguenze sono visibili. Confrontando i valori medi però è il protocollo Predictive ExtRef ad ottenere il valore migliore, risultato delle tecniche aggiuntive applicate al protocollo ExtRef. Malgrado l'introduzione del segnale GNSS invece il protocollo Qubit4Sync continua a comportarsi allo stesso modo non presentando nessuna visibile differenza, provando ancora una volta la sua robustezza.

Nelle Fig.[4.5][4.6], sono stati riportati i lag plot di primo ordine dei vari

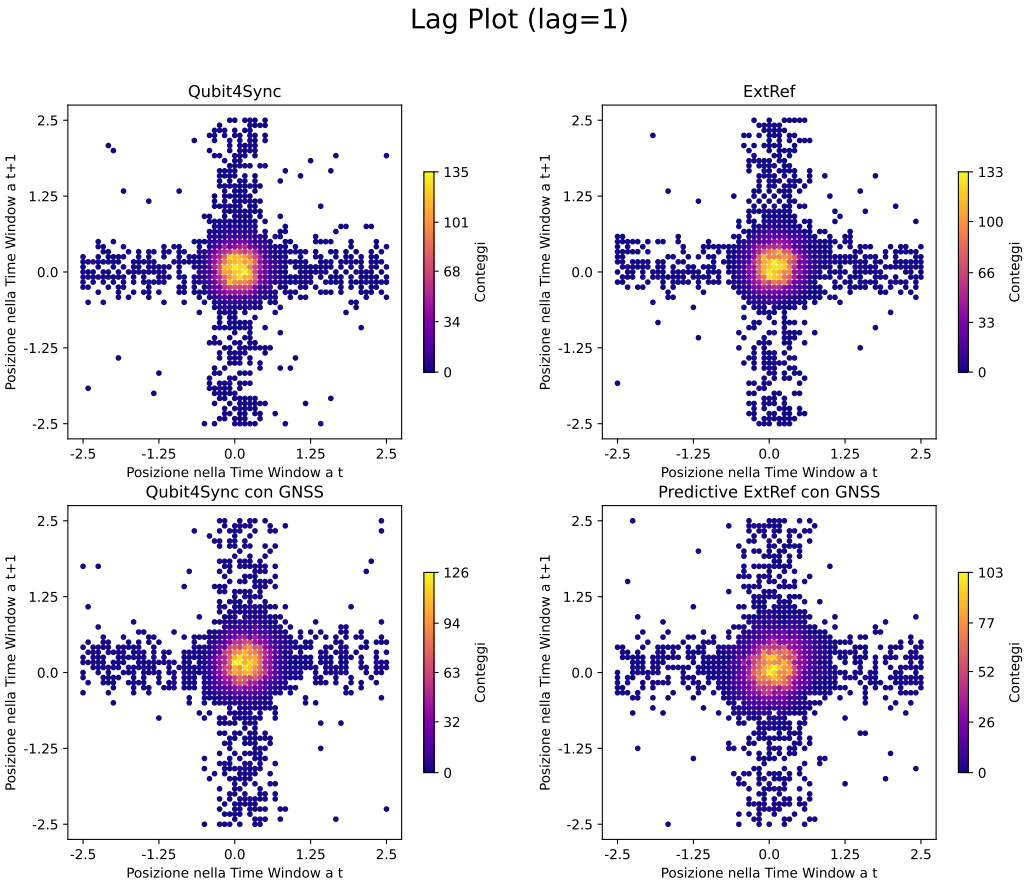


Figura 4.5: LagPlot 2D.

protocolli. Per riuscire ad avere un'idea più chiara della differenza di conteggi sono stati riportati i grafici utilizzando sia una rappresentazione bidimensionale che una tridimensionale, colorate con un'opportuna color map. In linea con quanto dedotto dalle figure precedenti, la più grande differenza e quella che si nota nel il protocollo Predictive ExtRef. Quest'ultimo infatti presenta una regione centrale, dove ricadono maggiormente gli impulsi, più ampia se paragonata a quelle degli altri protocolli. Questo fatto è conseguenza delle difficoltà di sincronia causate dai notevoli ed imprevedibili shift del segnale GNSS, per questo motivo gli impulsi si distribuiscono in un'area più ampia intaccando leggermente le performance del protocollo. Gli altri tre grafici sono molto simili e non si notano ulteriori differenze. Tutti e quattro i grafici infine presentano le due bande di punti, quella verticale e quella orizzontale. Come detto precedentemente queste bande indicano la presenza e la quantità di eventi singoli, originati dalle componenti rumorose sul canale quantistico,

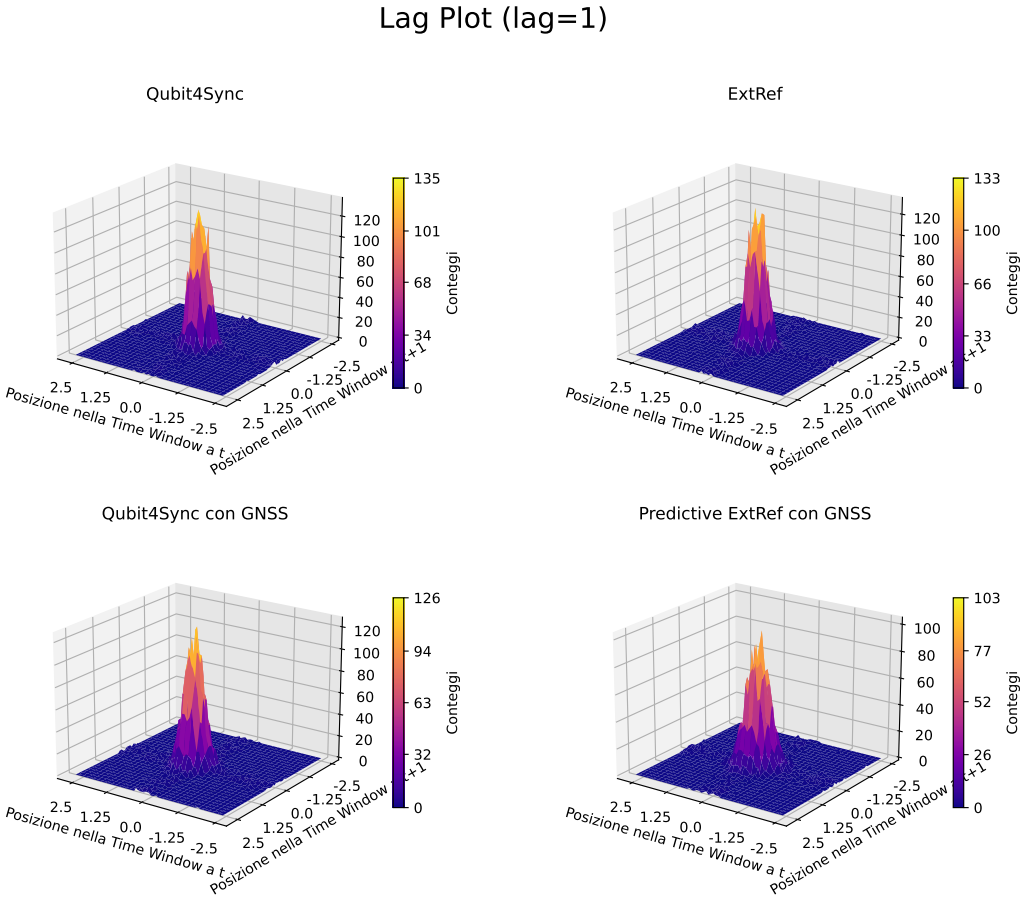
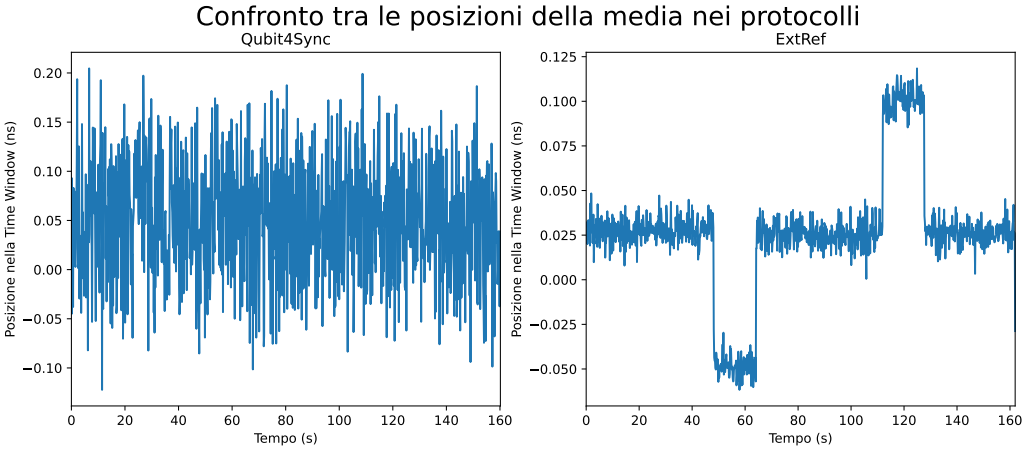


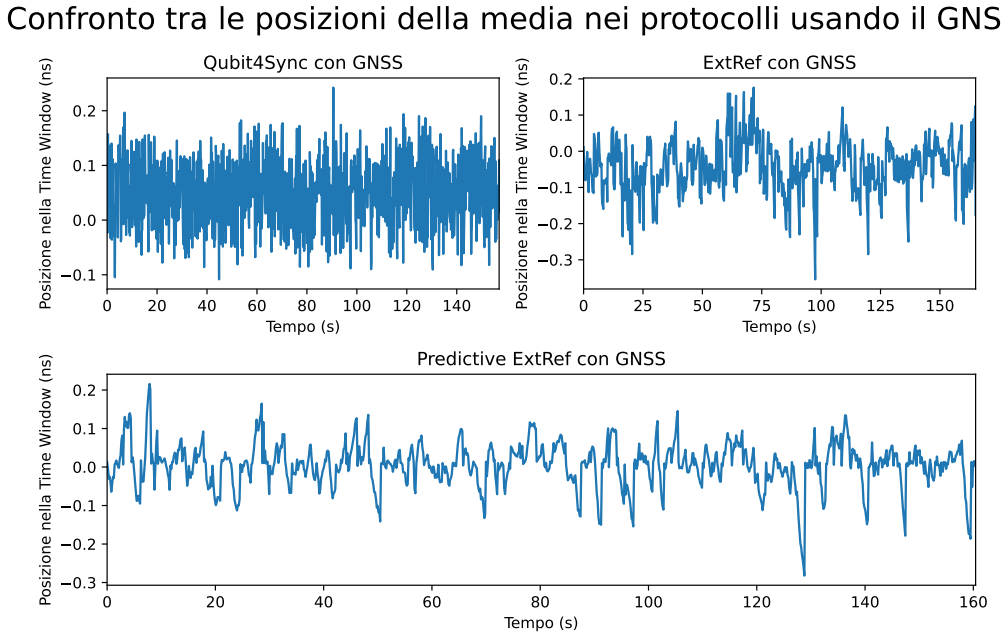
Figura 4.6: LagPlot3D.

che si trovano ad essere decentrati rispetto alla Time Window. Essi sono quelli che grazie all'utilizzo dei protocolli di sincronia vogliamo essere in grado di filtrare, abbassando in questo modo la possibilità di rilevazioni erronee. Infine questi grafici ci permettono di osservare come nei dati ricevuti non siano presenti evidenti dipendenze o errori sistematici evidenziando che i protocolli di sincronia lavorano correttamente non introducendo inesattezze.

Nelle Figure[4.7a][4.7b], per analizzare la stabilità dei diversi protocolli, è stato riportato il processo rappresentate il movimento della media dell'istogramma all'interno della Time Window. Per ottenere un confronto il più giusto possibile sono stati separati i casi in cui vengono utilizzati segnali di riferimento di qualità diversa. Questo genere di grafici ci è utile per valutare l'affidabilità e l'efficacia di un protocollo di sincronia.



(a) Confronto a lungo delle posizioni medie degli impulsi in uscita dai protocolli di sincronia.



(b) Confronto a lungo delle posizioni medie degli impulsi in uscita dai protocolli di sincronia che usano il GNSS.

Teoricamente, grazie al nostro protocollo di sincronia, vorremo far sì che la media del nostro istogramma fosse ferma al centro della Time Window, più il suo comportamento si allontana da questa situazione peggiore è il grado di sincronia raggiunto.

Com'è possibile osservare in Fig.[4.7a], il comportamento del protocollo Qubit4Sync e quello di ExtRef sono molto diversi. Nel primo il processo

rappresentante la posizione della media dell'istogramma oscilla in maniera continua attorno al valore del centro della Time Window senza mai fermarsi. Questa situazione è dovuta al fatto che il protocollo non avendo nessun riferimento esterno necessita di ripetere l'analisi, necessaria per restare centrato, ad ogni pacchetto, ricavando valori sempre leggermente differenti dello shift da applicare.

Come si può vedere invece, grazie al supporto del segnale di clock condiviso da Alice, il protocollo ExtRef ottiene risultati migliori da questo punto di vista, infatti grazie al segnale di riferimento può permettersi di aggiornare raramente il valore dello shift presente tra i due segnali (quello di riferimento e quello quantistico), ed in questo modo limita moltissimo le deviazioni del valore della media, che come è possibile osservare consistono solo di minime oscillazioni statistiche. Le due deviazioni dalla norma presenti in figura, sono dovute al fatto che negli istanti di aggiornamento del valore dello shift è stato analizzato un pacchetto sfortunato che si allontanava dal comportamento degli altri e questo ha causato che fino al prossimo aggiornamento dello shift il segnale risulti leggermente decentrato. Questi valori ad ogni modo sono più che accettabili se si pensa che la finestra di campionamento è larga un nanosecondo.

Nella Fig.[4.7b], viene analizzato il comportamento dei protocolli utilizzando come segnale di riferimento quello fornito dal sistema GNSS. Come è possibile osservare il comportamento del protocollo Qubit4Sync è rimasto praticamente inalterato. In questo caso l'introduzione del segnale GNSS non porta nessun beneficio a Bob che deve in ogni caso utilizzare soltanto l'informazione ricavabile dal canale quantistico per portare a termine la sincronizzazione. Tuttavia fornire ad Alice il segnale GNSS per disciplinare l'invio degli impulsi ottici dovrebbe ridurre i drift che altrimenti sarebbero presenti se quest'ultima utilizzasse il proprio clock interno. Ad ogni modo questo effetto non risulta visibile dai risultati presentati e il protocollo continua a mantenere lo stesso livello di precisione.

Per poter confrontare i risultati del protocollo ExtRef è stato necessario incrementare la frequenza con cui viene effettuato il controllo sulla fase del segnale quantistico, arrivando a far ripetere quest'analisi ogni 5 pacchetti ricevuti. Osservando il grafico si nota una notevole differenza con quello presentato in Fig.[4.7a]. Questo peggioramento è dovuto all'utilizzo del segnale GNSS, infatti ora anche se sia Alice che Bob ricevono un riferimento dal GNSS i segnali non sono perfettamente in fase, come dimostrato in Fig.[3.9], e le variazioni di quest'ultima causano oscillazioni nell'andamento della media dell'istogramma.

Infine il protocollo Predictive ExtRef, pensato per funzionare in situazioni dove persiste una variazione di fase tra i segnali, presenta un comportamen-

to simile a quello del protocollo ExtRef. In questo protocollo però essendo implementato un metodo predittivo in grado di applicare una correzione continua al segnale quantistico è possibile notare che anche se i picchi hanno la stessa estensione di quelli presenti nel protocollo ExtRef l'andamento generale è decisamente più piatto avvicinandosi di più al modello ideale, inoltre grazie all'algoritmo implementato in grado di ottimizzare il numero dei controlli necessari per mantenere la sincronia i vantaggi si protranno osservare anche analizzando il carico computazionale richiesto.

Terminando in Fig.[4.8] sono riportati i risultati di una più approfondita analisi del confronto tra il protocollo ExtRef e il protocollo Predictive ExtRef. Per poter confrontare questi due protocolli è stato usato come segnale di riferimento per entrambi il segnale GNSS. Per ottenere risultati confrontabili è stato necessario ridurre l'intervallo di aggiornamento dello shift del segnale quantistico del protocollo ExtRef, facendolo passare da ogni 100 pacchetti a ogni 5 pacchetti.

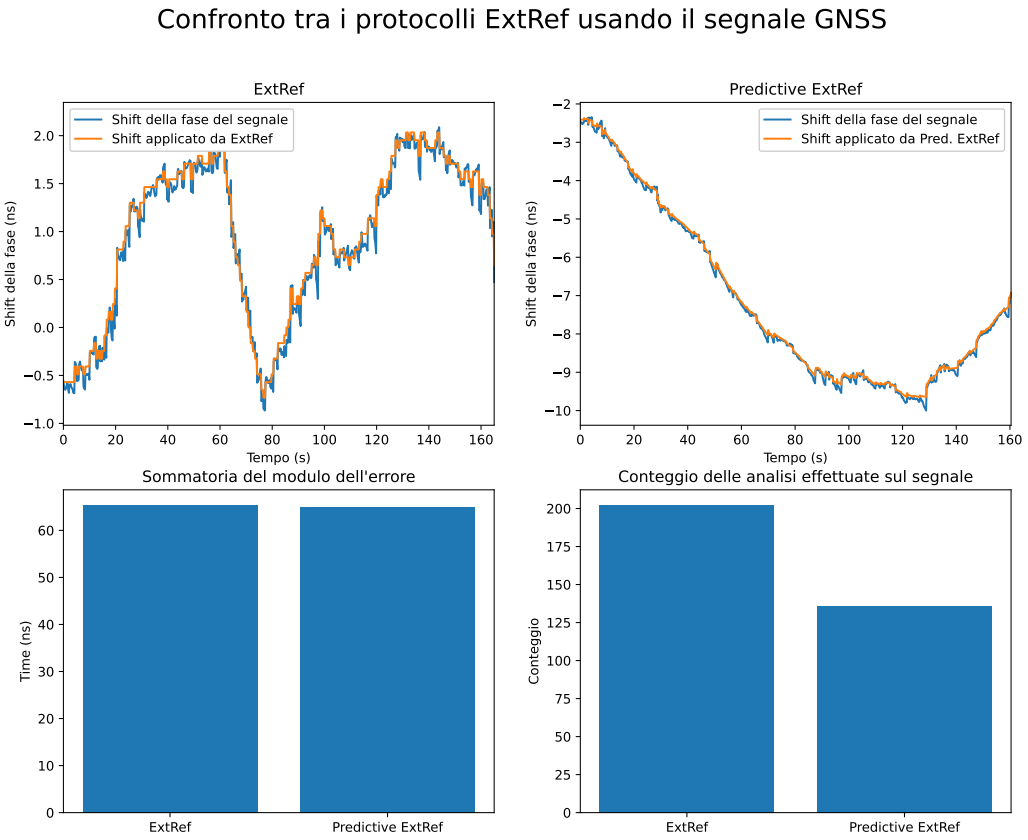


Figura 4.8: Confronto tra l'errore accumulato tra le due versioni di ExtRef.

Nei primi due grafici è possibile osservare come i due protocolli rispondono ai cambiamenti imprevedibili della fase del segnale quantistico. In questi due grafici è riportato l'effettivo cambiamento della fase del segnale quantistico e le correzioni applicate dai due protocolli per mantenere la fase con cui Bob riceve gli impulsi allineata a quella di Alice. Il protocollo ExtRef malgrado non applichi aggiustamenti continui del valore dello shift, come è possibile vedere dal grafico formato a scalini, riesce a restare vicino al valore esatto dello shift, questo grazie al fatto che la frequenza di aggiornamento di questo è stata aumentata. Il protocollo Predictive ExtRef con le sue correzioni resta aderente allo shift effettivo del segnale anche senza aver imposto nessun particolare vincolo. Grazie a questi due grafici è possibile valutare la reattività dei protocolli al cambiamento della fase del segnale quantistico che in entrambi i casi è molto buona. Come è possibile vedere dal terzo grafico, dove viene riportato l'integrale dell'errore commesso dai due protocolli, il livello raggiunto è praticamente lo stesso, mostrando come questo confronto sia stato effettuato cercando di porsi nelle condizioni il più possibile favorevoli per entrambi i protocolli.

Infine l'ultimo grafico mostra come per ottenere i precedenti risultati il protocollo Predictive ExtRef abbia ripetuto approssimativamente un terzo di volte in meno l'analisi sul segnale quantistico per determinare lo shift, evidenziando come questo algoritmo necessiti di un carico computazionale inferiore per ottenere li stessi risultati.

4.2 Hypothesis Testing

In questa sezione del capitolo dell'analisi dei risultati, vengono applicate tecniche statistiche di hypothesis testing per determinare se esiste una differenza significativa tra i risultati prodotti dai vari protocolli di sincronia presentati. Di norma per poter arrivare a delle conclusioni sarebbe necessario disporre dell'intera popolazione dei risultati ed esaminarne i parametri statistici. Essendo questo impossibile nel nostro caso, questi metodi permettono analizzando soltanto un campione limitato dei risultati, di verificare con un certo grado di affidabilità, se le assunzioni che facciamo possono essere considerate verosimili per l'intera distribuzione. Nel nostro caso siamo interessati a verificare quale delle due ipotesi H_0 o H_1 è valida.

H_0: *I protocolli di sincronia producono risultati equivalenti.*

H_1: *I protocolli di sincronia producono risultati che non sono equivalenti.*

Come è possibile notare le ipotesi da verificare sono l'una il negato dell'altra, in questo modo coprendo interamente lo spazio delle possibilità, si ha la certezza che una delle due venga verificata mentre l'altra sarà rifiutata. Una volta presentate le ipotesi è necessario specificare che queste tecniche sono soggette ad errori. Questi errori sono divisibili tipicamente in due categorie: gli errori di I e di II tipo. Gli errori di I tipo corrispondono alla probabilità di rifiutare l'ipotesi H_0 quando in realtà si rivela verificata, mentre gli errori di II tipo corrispondono alla probabilità di accettare l'ipotesi H_0 quando in realtà questa è falsa. In letteratura a questi due tipi di errori di vengono affiancate rispettivamente le variabili α e β. Dato che vogliamo determinare se esiste un'effettiva differenza tra i risultati prodotti dai protocolli, abbiamo interesse a minimizzare il parametro α. Valori tipici di accettazione usati in questo contesto, chiamati anche *significance level*, sono l'1% o il 5%, nel nostro caso abbiamo effettuato i test accettando valori inferiori al 5%. Così facendo abbiamo limitato la presenza di falsi positivi, ossia dell'eventualità in cui il test rifiuti H_0 quando in realtà l'ipotesi H_0 sarebbe verificata.

Una volta scelte le ipotesi da testare e la soglia di errore accettabile, è possibile confrontare i campioni ricavati dai nostri risultati con delle distribuzioni statistiche note e ricavare quello che in letteratura è noto come *p-value*. Questo valore ci permette di determinare qual è la probabilità di aver ottenuto la distribuzione dei risultati che stiamo analizzando assumendo che sia valida l'ipotesi H_0. Più è alto il valore del p-value meno probabile è che valga l'ipotesi H_0.

Nei nostri test abbiamo preso in esame la distribuzione *Student-t* e il test *ANOVA*.

4.2.1 Student-t Test

Lo Student-t test permette di valutare se le medie di due popolazioni differenti possono essere ricondotte alla stessa distribuzione. Per poter applicare questo test è necessario che valgano alcune condizioni.

1. **Relazione d'ordine:** Per i dati analizzati deve valere una relazione d'ordine, siano essi continui o discreti.

2. **Indipendenza dei campioni:** Ogni campione di dati deve essere indipendente dagli altri.

3. **Normalità:** Se rappresentato ogni campione di dati analizzato deve comportarsi approssimativamente come una distribuzione normale.

4. **Omoschedasticità:** La varianza dei dati dei campioni analizzati deve approssimativamente essere la stessa.

Nel nostro caso tutte le assunzioni sui dati vengono soddisfatte tranne l'ultima. Come abbiamo osservato in Fig.[4.4], esiste una differenza nella varianza tra i risultati prodotti dal protocollo Predictive ExtRef e gli altri. Fortunatamente anche se l'omoschedasticità rimane un suo prerequisito, lo Student-t test è stato dimostrato essere robusto alle variazione della varianza[24].

I dati raccolti per compiere questo test sono stati ottenuti ponendo ciascun protocollo nelle sue ideali condizioni di funzionamento. Nello specifico per il protocollo Qubit4Sync i clock utilizzati sono quelli interni di Alice e Bob, per il protocollo ExtRef si è utilizzato il clock di condiviso da Alice intervallando i controlli sulla fase del segnale quantistico ogni 100 pacchetti, infine per il protocollo Predictive ExtRef si è utilizzato il segnale di riferimento fornito dai ricevitori GNSS lasciando determinare la frequenza di controllo sulla fase al algoritmo slow-start. Questo test è stato applicato su tutte le possibili coppie di protocolli di sincronia presentati. Inoltre, per rendere più indicativi i risultati del test, lo abbiamo ripetuto per ogni coppia 100 volte, prendendo campioni di dati sempre differenti.

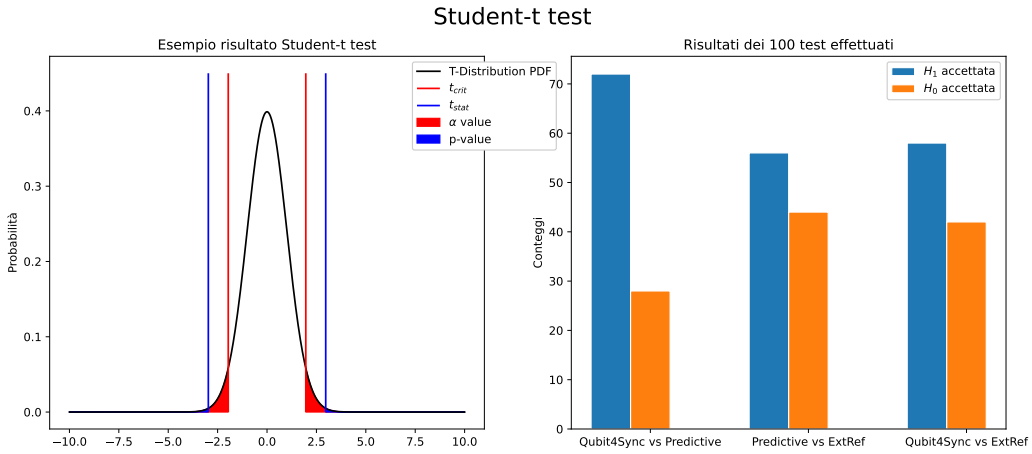


Figura 4.9: Rappresentazione del risultato dello Student-t test.

I risultati ottenuti sono rappresentati in Fig.[4.9]. Nel nostro caso dato che siamo interessati a capire se esiste una differenza significativa tra le medie dei due campioni, abbiamo implementato la variante *two-tailed* del test in quanto permette di verificare se la media della distribuzione di un campione è maggiore o minore dell'altra. Per poter svolgere lo Student-t test è necessario applicare ai dati raccolti la formula (4.1), dove $\bar{X}_1$ e $\bar{X}_2$ sono i valori medi dei campioni raccolti, s_p è la varianza aggregata, n è la dimensione dei campioni e s_{X_1} e s_{X_2} sono le varianze dei due campioni. Confrontando il valore t_{stat} ottenuto dai campioni con la sua distribuzione ideale, la *T-Distribution*,

è possibile quantificare la probabilità che questi appartengano dalla stessa distribuzione.

$$t_{stat} = \frac{\bar{X}_1 - \bar{X}_2}{s_p\sqrt{\frac{2}{n}}} \tag{4.1}$$

dove:

$$s_p = \sqrt{\frac{s^2_{X_1} + s^2_{X_2}}{2}}. \tag{4.2}$$

Nel primo grafico è stato riportato come esempio l'esito di un confronto tra la T-Distribution e l'effettiva statistica ottenuta dai campioni raccolti. Come si può osservare avere imposto un valore di accettazione di errore pari ad α permette di identificare nel grafico i valori soglia noti come t_{crit}. Se il valore ottenuto dalla statistica dei campioni t_{stat} si trova esternamente a questi valori allora vorrà dire che la distribuzione dei campioni differisce da quella attesa e il test terminerà rifiutando l'ipotesi H_0, mentre se t_{stat} dovesse trovarsi nella zona interna ai t_{crit} allora il test terminerà accettando l'ipotersi H_0. Osservando il parametro t_{stat} è inoltre possibile ricavare il valore numerico della probabilità che le due distribuzioni (quella nota e quella ottenuta dai campioni) siano simili. Questa probabilità si può ottenere integrando dalle code del grafico della T-Distribution fino alle soglie riportate, ed è comunemente nota come *p-value*.

Come si può notare dai dati del secondo grafico non sono stati trovati esiti decisivi. In ogni coppia di protocolli analizzata la percentuale di verifica dell'ipotesi H_0 non è trascurabile, invalidando in questo modo la possibilità che ne esista uno in grado di distinguersi nettamente dagli altri. Dati gli esiti del test accettiamo l'ipotesi H_0, affermando che i protocolli sono stati implementati tutti sufficientemente bene da produrre buone prestazioni.

4.2.2 ANOVA

Il test ANOVA (ANalysis Of VAriance), è un test statistico che permette, prendendo in esame un gruppo di campioni di popolazioni diverse, di determinare se queste appartengano alla stessa distribuzione. Anche per questo test valgono le gli stessi requisiti esposti per lo Student-t test.

Come per il test precedente anche qui, abbiamo ripetuto il confronto 100 volte. I risultati riportati in Fig.[4.10] vanno interpretati nella stessa maniera illustrata per lo Student-t test, con l'unica differenza che in questo caso ai dati è stata applicata la formula (4.3) per il calcolo di f_{stat}. La

formula (4.3) confronta il valore del *Mean Square for Treatment(MST)* con quello del *Mean Square for Error(MSE)*, il primo può essere pensato come un valore indicativo della varianza tra i K indipendenti campioni casuali mentre il secondo fornisce un'indicazione della varianza interna ai campioni. La statistica di confronto usata è la *F-Distribution*. I risultati estratti da questo test confermano quanto già osservato. Dato che il test non è stato in grado di identificare in maniera netta la validità dell'ipotesi H_1, i risultati ottenuti vanno interpretati come dimostrazione che non esiste un protocollo, tra quelli presentati, in grado di produrre risultati statisticamente differenti. Tutte e quattro le varianti quindi, applicate ai rispettivi scenari operativi, sono in grado di raggiungere ottimi livelli di funzionamento.

$$f_{stat} = \frac{\text{MST}}{\text{MSE}} \tag{4.3}$$

dove:

$$n = \sum_{k=1}^{K} n_k \tag{4.4}$$

$$MST = \frac{\sum_{k=1}^{K} n_k(\bar{X}_k - \bar{X})}{K-1} \tag{4.5}$$

$$MSE = \frac{\sum_{k=1}^{K}(n_k - 1)s_k^2}{n-K} \tag{4.6}$$

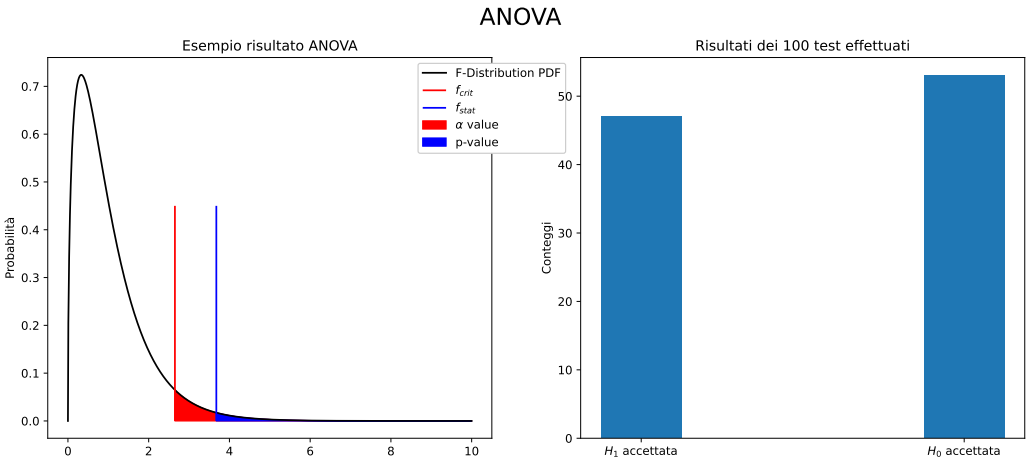


Figura 4.10: Rappresentazione del risultato del test ANOVA.

In questo capitolo di analisi dei risultati siamo andati a fondo analizzando i comportamenti dei vari protocolli e abbiamo capito che ciò che li differenzia di più non sono i risultati che producono, tutti generalmente molto buoni, ma le risorse che richiedono. Infatti ad esempio, il protocollo Qubit4Sync è molto flessibile e può essere applicato anche in situazione per le quali non è stato pensato, però richiede che ogni pacchetto del segnale quantistico venga analizzato per mantenere l'allineamento tra i dispositivi. Il protocollo ExtRef riesce a ridurre il carico computazionale ma necessita che il segnale di clock di Alice venga condiviso con Bob. Infine il protocollo Predictive ExtRef riesce a ridurre il carico computazionale ma per poter ricevere il segnale dal GNSS ha bisogno di ricevitori GNSS. Ognuno di questi protocolli si rivela adatto per instaurare la sincronia tra Alice e Bob, il fattore discriminante sono le condizioni di applicazione.

4.3 Funzionamento in scenari critici

Per poter valutare la resistenza dei protocolli ai disturbi sul canale quantistico è stato progettato un sistema che permette di simulare l'effetto di un disturbo. Come è possibile osservare in Fig.[4.11], tramite l'introduzione di un *Intensity Modulator*[1] sul canale quantistico è possibile far variare in maniera dinamica l'attenuazione, impedendo che il segnale riesca ad attraversare il collegamento ottico. Con l'intento di simulare un possibile scenario di applicazione dove il segnale quantistico viene tramesso su collegamenti free space, le frequenze dei disturbi sono state impostate per variare tra i $10Hz$ e i $100Hz$. Fornendo in ingresso all'IM un clock con queste frequenze il segnale quantistico verrà disturbato. Un esempio del segnale risultante che raggiunge Bob è stato riportato in Fig.[4.12].

Come è possibile osservare, l'effetto dell'IM si è tradotto in periodiche mancanze di segnale. Purtroppo utilizzando questo modello di canale disturbato non siamo riusciti a portare a termine l'esecuzione del protocollo. Tutti i test effettuati hanno manifestato lo stesso comportamento, malgrado la presenza del disturbo Alice e Bob sono riusciti ad allineare le proprie basi ma in nessun caso l'esecuzione ha superato l'analisi della stringa di sincronia raggiungendo lo stato di trasmissione della chiave grezza. Analizzando i messaggi di log prodotti è stato notato che l'esecuzione dei protocolli terminava sempre nello stesso punto, appena dopo aver calcolato la cross-correlazione tra il segnale ricevuto da Bob e la chiave di sincronia, fallendo un controllo sul valore del massimo calcolato da quest'ultima. Questo controllo è stato

[1] Un *Modulatore di Intensità* è un dispositivo elettronico pilotato da un segnale elettrico che permette di modulare l'intensità di un segnale ottico che lo attraversa.

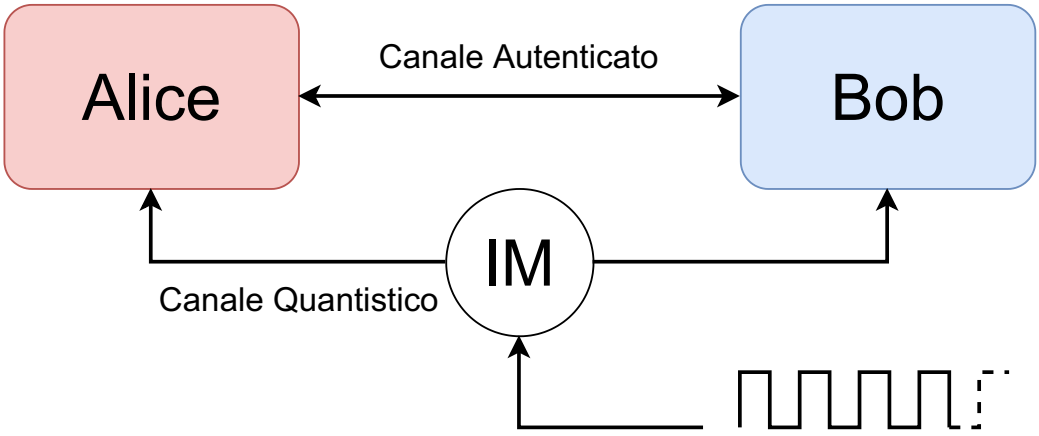


Figura 4.11: Rappresentazione della configurazione usate per i test negli scenari critici.

inserito nel codice del protocollo per accertarsi che in casi di canali particolarmente rumorosi la sincronia non si agganci su valori sbagliati. In questo caso però il valore del massimo trovato dalla cross-correlazione è stato intaccato per la presenza dei buchi nel segnale quantistico. Infatti se si considera la lunghezza della stringa di sincronia utilizzata ossia $10Mbit$ e la frequenza di trasmissione di Alice cioè $50MHz$ si può concludere che l'intervallo di tempo necessario alla trasmissione della stringa è di $1/5$ di secondo. Essendo necessario un intervallo temporale così lungo per la trasmissione dell'intera stringa di sincronia è inevitabile che i disturbi che abbiamo simulato sul canale incidano sulla sua ricezione. A causa di questo spiacevole comportamento il segnale ricevuto da Bob contenente la stringa di sincronia non raggiunge mai il valore minimo richiesto dal massimo della cross-correlazione per superare questo passaggio.

Il modello di canale disturbato realizzato si è rivelato troppo aggressivo andando ad impedire la terminazione dei protocolli di sincronia. Nonostante questo però osservando la Fig.[4.12] rappresentante, come esempio, i dati raccolti durante l'esecuzione del protocollo ExtRef in condizioni di canale disturbato, è possibile concludere che la condivisione di periodo e fase tra Alice e Bob è comunque stata raggiunta. Infatti osservando il grafico dell'istogramma[2] è possibile notare la presenza della gaussiana e il fatto che

[2]Per la raccolta dei dati nella condizione di canale disturbato è stato necessario cambiare coppia di sistemi Alice e Bob. Utilizzando una configurazione diversa è stato riscontrata una maggiore quantità di rumore sul canale quantistico, come è possibile osservare notando il plateau dell'istogramma più accentuato. Questo in ogni caso non altera le conclusioni dell'analisi.

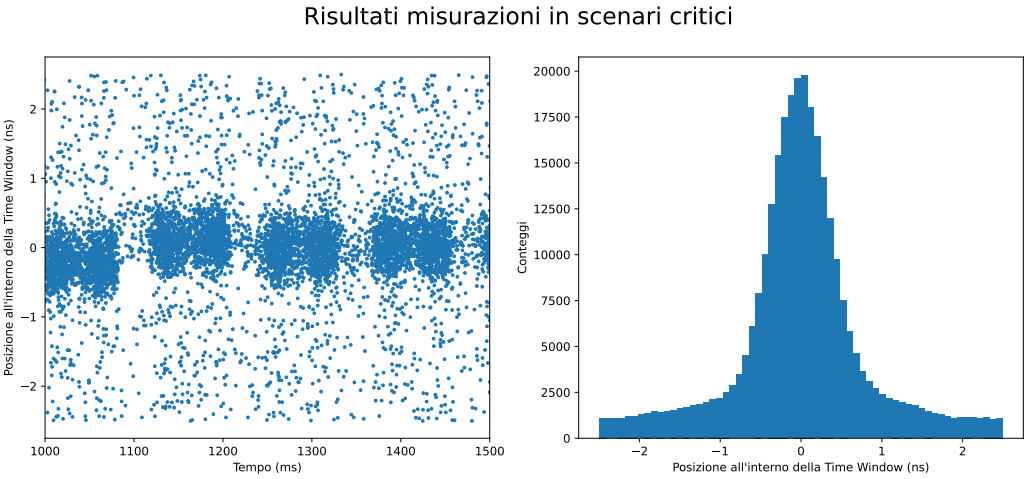


Figura 4.12: Segnale quantistico disturbato dal IM.

questa sia perfettamente centrata sullo zero della Time Window. Da queste osservazioni è possibile concludere che il calcolo del periodo del segnale quantistico e l'aggiustamento della fase sono entrambe procedure portate correttamente a completamento. Alcuni cambiamenti attuabili per poter testare più efficacemente la sincronia in condizioni di canale disturbato potrebbero essere quello di utilizzare stringhe di sincronia di lunghezza diversa oppure quello di studiare se sono accettabili valori più bassi per il massimo della cross-correlazione.

Capitolo 5

Conclusioni

In questa tesi ho esposto l'esperienza che ho avuto lavorando con il gruppo QuantumFuture all'interno dei laboratori di Padova, dove ho potuto partecipare alla realizzazione delle tecniche crittografiche di quantum key distribution. Concentrandomi sullo studio e l'analisi dei protocolli di sincronia ho provato a valutare le caratteristiche che li contraddistinguono determinando a quali esigenze rispondessero e in quali scenari potessero trovare maggiore applicabilità. La possibilità di avere accesso a risorse differenti mi ha permesso di realizzare protocolli in grado di approcciare il problema da un diverso punto di vista, non più incentrato sull'elaborazione e l'analisi, ma rivolto maggiormente alla riduzione dell'impiego delle risorse computazionali, ed anche se c'è stato bisogno di accettare qualche compromesso i risultati ottenuti sono stati soddisfacenti.

Come si ha avuto modo di considerare dalla lettura di questa tesi, la crittografia quantistica è un ambito di ricerca particolarmente attivo in quanto punta a proporsi come successore degli odierni metodi crittografici. Nonostante le sue fondamenta siano state ampiamente dimostrate e i risultati che è in grado di offrire osservati, prima della commercializzazione di questa tecnologia è ancora necessario un più estensivo sviluppo della rete ottica sul territorio e la realizzazione di soluzioni in grado di scalare in base ai bisogni sempre crescenti delle persone di scambiare informazioni in sicurezza.

Appendice A

Gestione del progetto

In questo capitolo vengono descritti gli strumenti utilizzati per la gestione del progetto, discutendo le motivazioni che hanno spinto ad adottare certe scelte per il design e lo sviluppo del codice.

Realizzare un protocollo di comunicazione da zero non facile, tutto quello che è trattato all'interno di questa tesi riguarda solamente l'aspetto di sincronizzazione dei dispositivi, non affrontando le ulteriori problematiche presenti per far funzionare assieme tutte le apparecchiature necessarie.

Per progredire efficacemente e facilitare lo sviluppo del codice del progetto è stato scelto di mettere in primo piano le caratteristiche di facilità di accesso, automatizzazione e modularità del codice. Queste proprietà sono fondamentali per la produzione di codice di qualità e sono gli stessi principi seguiti dalle figure professionali del settore.

- **Facilità di accesso:** La possibilità di poter accedere alle risorse del progetto in modo semplice e veloce da qualsiasi dispositivo è una caratteristica fondamentale che rende il progetto più flessibile e resistente ai cambiamenti che possono presentarsi durante il suo sviluppo.

- **Automatizzazione:** Riuscire a gestire centinaia di file non è semplice e pensare di doverlo ripetere ogni volta che è necessario riconfigurare la build del progetto non è proponibile. Per questo motivo è necessario automatizzare le operazioni più lunghe e tediose ma necessarie come ad esempio il download ed il linking delle dipendenze e librerie necessarie al progetto per funzionare.

- **Modularità:** La capacità di riuscire a separare lo sviluppo delle diverse componenti del progetto è fondamentale per poterne facilitare la realizzazione. Ogni componente all'interno del progetto deve avere una funzione propria che la caratterizza, in questo modo è possibile creare

una struttura più ordinata del codice ed individuare più facilmente gli errori.

A.1 Controllo versione

GitLab e GitKraken sono due applicativi basati su Git in grado di sinergizzare molto bene e semplificare di molto lo sviluppo di progetti software di qualsiasi dimensione. Il sistema di controllo versione distribuito realizzato da Git è uno degli strumenti più utilizzati per la realizzazione di software moderno il cui sviluppo è diviso tra più persone. Questo sistema permette di creare una vera e propria cronologia dello sviluppo del codice del progetto, consentendo di muoversi senza impedimenti tra i vari commit, libera gli sviluppatori dalla paura di sbagliare, permettendogli in ogni momento di ripristinare la versione precedente del codice e ricominciare.

Un'altra utilissima funzionalità di Git consiste nella possibilità di creare varie branch dello stesso codice. Quando si vuole implementare una nuova funzionalità, senza compromettere l'integrità del progetto stesso, è possibile diramarne lo sviluppo rendendo le due versioni indipendenti l'una dall'altra. Una volta terminato lo sviluppo delle funzionalità aggiuntive è possibile far convergere nuovamente i flussi di lavoro formando una nuova versione che comprenda il codice aggiuntivo. Utilizzando lo stesso meccanismo delle branch è possibile far lavorare allo stesso codice vari sviluppatori in contemporanea, evitando che il lavoro di uno entri in conflitto con quello degli altri.

Nello sviluppo dei vari protocolli di sincronia tutte queste funzioni sono state ampiamente utilizzate, riuscendo a garantire uno sviluppo pratico e agevole del progetto senza rallentare il progresso dei numerosi lavori in corso all'interno dei laboratori.

Scendendo nel dettaglio, anche se GitLab e GitKraken sono basati sullo stesso sistema di controllo versione rispondono ad esigenze diverse. GitLab è un servizio di archiviazione cloud, che permette di salvare in sicurezza su un server tutto l'albero dei commit dello sviluppo del software, rendendo accessibile il codice ovunque. Mentre GitKraken è un editor grafico che permette di visualizzare lo sviluppo del codice e grazie alla sua interfaccia grafica semplifica l'applicazione delle funzionalità di Git.

(a) Logo di GitLab

(b) Logo di GitKraken

Figura A.1: Loghi dei software di controllo versione utilizzati nel progetto.

A.2 Build system

CMake è uno strumento open source multipiattaforma per l'automazione dello sviluppo. Quando si ha a che fare con progetti particolarmente grandi è facile che le dipendenze e le librerie a cui facciamo riferimento siano molto più corpose e numerose dello stesso codice che stiamo scrivendo. Per poter istruire il sistema su come deve essere compilato il nostro progetto è possibile servirsi di CMake e creare un file di configurazione in grado di collegare correttamente tutti i file coinvolti, semplificando così il processo di compilazione.

Durante lo sviluppo del codice la compilazione è un'operazione fondamentale che viene ripetuta innumerevoli volte, avere un applicativo che dietro le quinte predisponga tutto il necessario perché questa sia efficiente e vada a buon fine è estremamente utile. Inoltre cambiare sistema dove si sta sviluppando il codice in corso d'opera, è una cosa comune che può succedere. Nello sviluppo dei protocolli di sincronia è successo più volte che le macchine dove si stava testando il codice da un giorno all'altro non fossero più disponibili. Grazie a agli automatismi offerti da CMake ripristinare la build del progetto è stato estremamente facile e rapido.

Figura A.2: Logo di CMake

A.3 Programmazione ad eventi

Per sviluppare l'intero progetto e quindi anche i protocolli di sincronia è stata utilizzata la libreria multipiattaforma Qt. Questa libreria si appoggia al linguaggio C++ e lo estende semplificandone vari aspetti.

Il linguaggio C++ è stato scelto dai componenti del gruppo QuantumFuture in quanto estremamente efficiente ed in grado di interfacciarsi direttamente con la macchina senza bisogno di creare nessun layer di astrazione. Lo svantaggio di aver scelto questo linguaggio è che non dispone di molte funzioni e librerie che invece sono presenti di default in linguaggi più astratti, un esempio è la gestione della memoria, completamente demandata allo sviluppatore.

Il paradigma di programmazione utilizzato nel progetto è quello dello sviluppo di applicazioni ad eventi. Questa tecnica di programmazione permette di controllare il flusso d'esecuzione del codice consentendo di interagire con l'esecuzione delle istruzioni, in questo modo è possibile gestire lo svolgimento del protocollo da un'interfaccia grafica durante il runtime. Grazie a Qt realizzare interfacce grafiche, relazionarsi con database SQL, effettuare il parsing di documenti XML sono operazioni estremamente facili da effettuare. All'interno del codice dei protocolli di sincronia le funzionalità di Qt sono state utili anche per creare collegamenti tra i socket dei dispositivi attraverso la rete, permettendo ad Alice e Bob di scambiarsi informazioni tramite rete ethernet. Il meccanismo che contraddistingue Qt è quello dei *signal* e degli *slot*. Con il loro funzionamento è stato possibile realizzare del codice multithread in grado di suddividere il carico computazionale tra processi diversi. Anche se non risulta essere particolarmente innovativo, il sistema di Qt di effettuare chiamate di funzione passando gli argomenti tra processi diversi, è una caratteristica molto utile implementata anche nel progetto del team QuantumFuture.

Figura A.3: Logo di Qt

Bibliografia

[1] Alessandro Languasco e Alessandro Zaccagnini. *Manuale di crittografia: teoria, algoritmi e protocolli*. Hoepli Editore, 2020.

[2] Andrew Hodges. «Alan Turing: the enigma». In: *Alan Turing: The Enigma*. Princeton University Press, 2014.

[3] Ronald L Rivest, Adi Shamir e Leonard Adleman. «A method for obtaining digital signatures and public-key cryptosystems». In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[4] P.W. Shor. «Algorithms for quantum computation: discrete logarithms and factoring». In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[5] Wikipedia contributors. *One-time pad — Wikipedia, The Free Encyclopedia*. 2022. URL: https://en.wikipedia.org/w/index.php?title=One-time_pad&oldid=1091714627.

[6] C. E. Shannon. «Communication theory of secrecy systems». In: *The Bell System Technical Journal* 28.4 (1949), pp. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.

[7] William K Wootters e Wojciech H Zurek. «A single quantum cannot be cloned». In: *Nature* 299.5886 (1982), pp. 802–803.

[8] Andrea Albino et al. «First-principles investigation of spin–phonon coupling in vanadium-based molecular spin quantum bits». In: *Inorganic chemistry* 58.15 (2019), pp. 10260–10268.

[9] Govind P Agrawal. *Fiber-optic communication systems*. John Wiley & Sons, 2012.

[10] Mohammad Ali Khalighi e Murat Uysal. «Survey on free space optical communication: A communication theory perspective». In: *IEEE communications surveys & tutorials* 16.4 (2014), pp. 2231–2258.

[11] Bing Qi, Li Qian e Hoi-Kwong Lo. *A brief introduction of quantum cryptography for engineers*. 2010. arXiv: 1002.1237 [quant-ph].

[12] Charles H Bennett e Gilles Brassard. «Quantum cryptography: Public key distribution and coin tossing». In: *arXiv preprint arXiv:2003.06557* (2020).

[13] Valerio Scarani et al. «The security of practical quantum key distribution». In: *Rev. Mod. Phys.* 81 (3 set. 2009), pp. 1301–1350. DOI: 10.1103/RevModPhys.81.1301. URL: https://link.aps.org/doi/10.1103/RevModPhys.81.1301.

[14] Hoi-Kwong Lo, Hoi Fung Chau e Mohammed Ardehali. «Efficient quantum key distribution scheme and a proof of its unconditional security». In: *Journal of Cryptology* 18.2 (2005), pp. 133–165.

[15] Kiyoshi Tamaki et al. «Loss-tolerant quantum cryptography with imperfect sources». In: *Physical Review A* 90.5 (2014), p. 052314.

[16] Charles H Bennett. «Quantum cryptography using any two nonorthogonal states». In: *Physical review letters* 68.21 (1992), p. 3121.

[17] Charles H Bennett, Gilles Brassard e N David Mermin. «Quantum cryptography without Bell's theorem». In: *Physical review letters* 68.5 (1992), p. 557.

[18] Costantino Agnesi et al. «All-fiber self-compensating polarization encoder for quantum key distribution». In: *Opt. Lett.* 44.10 (mag. 2019), pp. 2398–2401. DOI: 10.1364/OL.44.002398. URL: http://opg.optica.org/ol/abstract.cfm?URI=ol-44-10-2398.

[19] Luca Calderaro et al. «Fast and Simple Qubit-Based Synchronization for Quantum Key Distribution». In: *Phys. Rev. Applied* 13 (5 mag. 2020), p. 054041. DOI: 10.1103/PhysRevApplied.13.054041. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.13.054041.

[20] Lei M. Li. «An algorithm for computing exact least-trimmed squares estimate of simple linear regression with constraints». In: *Computational Statistics & Data Analysis* 48.4 (2005), pp. 717–734. ISSN: 0167-9473. DOI: https://doi.org/10.1016/j.csda.2004.04.003. URL: https://www.sciencedirect.com/science/article/pii/S0167947304001082.

[21] Behrouz A Forouzan. *TCP/IP protocol suite*. McGraw-Hill Higher Education, 2002.

[22] David L Mills. «Internet time synchronization: the network time protocol». In: *IEEE Transactions on communications* 39.10 (1991), pp. 1482–1493.

[23] Kendall Correll, Nick Barendt e Michael Branicky. «Design considerations for software only implementations of the IEEE 1588 precision time protocol». In: *Conference on IEEE*. Vol. 1588. Citeseer. 2005, pp. 11–15.

[24] Carol A. Markowski e Edward P. Markowski. «Conditions for the Effectiveness of a Preliminary Test of Variance». In: *The American Statistician* 44.4 (1990), pp. 322–326. ISSN: 00031305. URL: http://www.jstor.org/stable/2684360 (visitato il 29/06/2022).