

STUDIO SIMULATIVO E SVILUPPO DI PROTOCOLLI
DI ACCESSO AL MEZZO PER RETI ACUSTICHE
SOTTOMARINE

RELATORE: Ch.mo Prof. Michele Zorzi

CORRELATORE: Dott. Ing. Paolo Casari

LAUREANDO: Federico Favaro

A.A. 2011-2012



UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
TESI DI LAUREA

STUDIO SIMULATIVO E SVILUPPO
DI PROTOCOLLI DI ACCESSO AL
MEZZO PER RETI ACUSTICHE
SOTTOMARINE

RELATORE: Ch.mo Prof. Michele Zorzi

CORRELATORE: Dott. Ing. Paolo Casari

LAUREANDO: *Federico Favaro*

Padova, 12 Dicembre 2011

A mamma e papà

A Chiara

Indice

Sommario	1
1 Una panoramica sulle comunicazioni acustiche sottomarine	2
1.1 Caratteristiche della propagazione e del canale acustico sottomarino	2
1.2 Caratteristiche di livello fisico	11
2 I protocolli di accesso al mezzo per le comunicazioni underwater	14
2.1 CSMA-ALOHA	15
2.2 DACAP	17
2.3 TONE-LOHI	22
2.3.1 Synchronized TONE-LOHI (ST-Lohi)	23
2.3.2 Conservative Unsynchronized TONE-LOHI (cUT-Lohi) . .	23
2.3.3 Aggressive Unsynchronized TONE-LOHI (aUT-Lohi) . . .	23
2.4 USR-AIMD	24
3 Simulazioni e risultati	27
3.1 Primo scenario : nodi in posizione casuale	31
3.1.1 Shallow-water	31
3.1.2 Deep-water	45
3.2 Secondo scenario: nodi in posizione fissa che comunicano con un sink superficiale	55
3.2.1 Shallow-water	55
3.2.2 Deep-water	63
3.3 Conclusioni e commenti	72
4 UW-POLLING	73
4.1 Il protocollo UW-POLLING (Nodo)	75

INDICE

4.2	Il protocollo UW-POLLING (AUV)	76
4.3	Scenario di simulazione e risultati	78
4.3.1	Zona di simulazione e posizionamento dei nodi	78
4.3.2	Risultati delle simulazioni	79
4.4	Conclusioni e commenti	90
A	Bellhop ray-tracing	92
B	Ns-Miracle e WOSS	97
	Bibliografia	99

Sommario

Le comunicazioni acustiche sottomarine sono state oggetto di numerosi studi, con vari impieghi, dal monitoraggio di costruzioni sottomarine (stazioni di perforazione e pipeline di petrolio), alla sicurezza militare e al monitoraggio di valori ambientali tipici del mare.

L'esigenza di una comunicazione efficiente in ambito underwater, solleva alcuni problemi nuovi per le telecomunicazioni. In particolare, i protocolli di accesso al mezzo (MAC - *Medium Access Control*), devono essere rivisti, in quanto i tempi di propagazione diventano considerevoli se paragonati alle comunicazioni radio terrestri. (Il suono si propaga in acqua a una velocità tipica di 1500 m/s). I suddetti protocolli devono essere rivisti anche alla luce delle nuove caratteristiche del link fisico, caratterizzato da una banda molto stretta (tipicamente di pochi kHz), da elevata latenza e effetti di fading da cammini multipli spesso molto sfavorevoli.

Utilizzando il motore di simulazione *WOSS* [2], è possibile simulare in maniera realistica il comportamento dei protocolli in presenza di un canale acustico sottomarino aderente alla realtà utilizzando Network Simulator [12] accoppiato a NSMiracle [8]. Il software Bellhop, tramite *ray-tracing*, si propone di fornire delle informazioni sulla propagazione acustica da cui si può ricavare, per esempio, la attenuazione del canale, sulla base dei dati ricavati dall'interrogazione dei databases contenenti i dati ambientali con maggiore impatto sulla propagazione acustica.

Questa tesi si propone, utilizzando questi strumenti di simulazione, di testare in ambienti diversi, alcuni protocolli MAC esistenti, cercando di evidenziare quale protocollo e quale politica di trasmissione forniscono prestazioni migliori in termini di throughput, Packet Delivery Ratio e altre metriche di rete.

Capitolo 1

Una panoramica sulle comunicazioni acustiche sottomarine

1.1 Caratteristiche della propagazione e del canale acustico sottomarino

Mentre la ricerca è estremamente attiva nella ricerca della capacità del canale radio wireless, non esistono ancora caratterizzazioni “universali” del canale acustico sottomarino.

Con la repentina ascesa dell’interesse per le comunicazioni underwater, si è compresa ben presto la necessità di avere un adeguato modello della propagazione nell’ambiente marino.

Le comunicazioni acustiche sottomarine sono affette da un *path-loss* dipendente non solo dalla distanza tra trasmettitore e ricevitore, ma anche dalla frequenza. [4] Questa dipendenza è dovuta a una perdita per assorbimento a causa della trasformazione di energia acustica in calore dovuta agli urti atomici. In questo senso, quindi, la frequenza portante delle trasmissioni nella nostra rete ha un impatto significativo sull’attenuazione a cui è sottoposto il nostro segnale.

Definiamo l’attenuazione per un segnale di frequenza f (in kHz) che percorre una distanza l (in metri) come:

$$A(l, f) = l^k a(f)^l \quad (1.1)$$

la cui espressione in dB diventa quindi

$$10 \log A(l, f) = k \cdot 10 \log l + l \cdot 10 \log a(f) \quad (1.2)$$

dove il primo termine della somma rappresenta lo *spreading loss* e il secondo rappresenta l' *absorption loss*. Il termine k rappresenta la geometria della propagazione.

A titolo di esempio, in un ambiente deep-water, in cui la propagazione del segnale è libera da “ostacoli” naturali come riflessioni nella superficie o nel fondale si utilizza tipicamente $k = 2$ che rappresenta una propagazione sferica, mentre in un ambiente shallow-water, dove le riflessioni sono più probabili il valore più corretto da utilizzare è $k = 1$ che rappresenta una propagazione cilindrica.

Il coefficiente $a(f)$ (chiamato coefficiente di assorbimento) è ottenuto tramite formule empiriche chiamate *formule di Thorp* o anche *formule di Urick*:

$$10 \log a(f) = \frac{1}{1000} \left(0.11 \frac{f^2}{1 + f^2} + 44 \frac{f^2}{4100 + f^2} + 2.75 \cdot 10^{-4} f^2 + 0.003 \right) \quad (1.3)$$

La formula (1.3) è valida per frequenze superiori a un centinaio di Hz. Per valori di frequenza sotto a questa soglia si utilizza invece

$$10 \log a(f) = \frac{1}{1000} (0.002 + 0.11 \frac{f^2}{1 + f^2} + 0.11 f^2) \quad (1.4)$$

dove la frequenza è sempre espressa in kHz.

La Figura (1.1) riporta un grafico di $a(f)$ al variare della frequenza; possiamo, in particolare, vedere come l'attenuazione cresca molto velocemente all'aumentare della frequenza di lavoro. Queste formule rappresentano il modello più semplice del canale che ci permette di eseguire delle simulazioni di tentativo per avere, almeno inizialmente, una panoramica sulle prestazioni dei protocolli da testare.

Per avere un modello più fedele della realtà in futuro verranno utilizzati WOSS [2] e Bellhop [13]. In Figura (1.4) possiamo vedere l'andamento del fattore $A(l, f)$ in funzione della frequenza, per alcune distanze campione. Il grafico mostra la dipendenza del path loss dalla frequenza.

Per quanto riguarda il rumore, in ambiente in underwater abbiamo quattro sorgenti principali di rumore: *turbolenza*, *navigazione*, *onde* e *rumore termico*.

Come si può intuire facilmente dai nomi, i primi tre sono proprio tipici dell'ambiente sottomarino, mentre il rumore termico è conosciuto anche nei canali radio terrestri. Le seguenti sono formule empiriche che danno la densità spettrale di

1. UNA PANORAMICA SULLE COMUNICAZIONI ACUSTICHE SOTTOMARINE

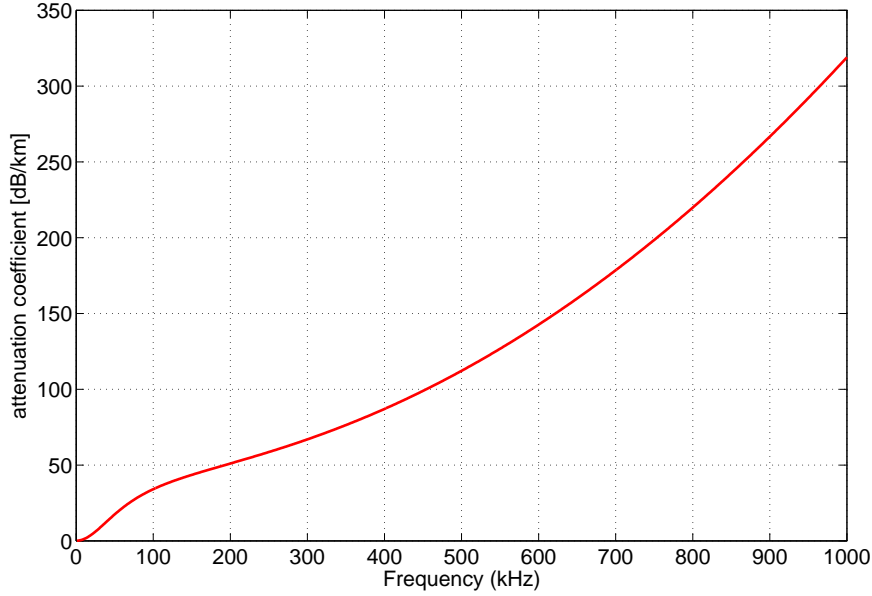


Figura 1.1: Coefficiente di attenuazione $a(f)$ al variare della frequenza.

potenza delle componenti in dB *re* μPa per Hz in funzione della frequenza di lavoro in kHz [4] :

$$10 \log N_t(f) = 17 - 30 \log f \quad (1.5)$$

$$10 \log N_s(f) = 40 + 20(s - 0.5) + 26 \log f - 60 \log(f + 0.03) \quad (1.6)$$

$$10 \log N_t(f) = 50 + 7.5w^{1/2} + 20 \log f - 40 \log(f + 0.4) \quad (1.7)$$

$$10 \log N_t(f) = -15 + 20 \log f \quad (1.8)$$

Il rumore dovuto alla turbolenza è dato dalla formula (1.5) ed è dominante per frequenze al di sotto dei 10 Hz. Per quanto riguarda la navigazione, il contributo è dato dalla formula (1.6); questo contributo diventa significativo nella regione di frequenze dai 10 Hz ai 100 Hz. Il fattore s indica il livello di navigazione e ha un range da 0 (navigazione assente) a 1 (traffico marittimo molto elevato).

Il contributo delle onde è dovuto al vento (il fattore w è la velocità del vento in m/s) ed è dato dalla formula (1.7). Questa componente di rumore è dominante nel range di frequenze 100 Hz - 100 kHz, nel range di frequenze dove lavora la maggior parte delle reti acustiche sottomarine.

Il rumore termico invece, a differenza delle reti wireless terrestri, fornisce un contributo più modesto nelle reti underwater, perchè diventa significativo a frequenze

maggiori di 100 kHz; frequenze che cominciano a essere troppo elevate per applicazioni acustiche sottomarine. In Figura (1.2) possiamo vedere l'insieme di tutti i contributi di rumore al variare della frequenza. In Figura (1.3), invece, possiamo vedere l'effetto simultaneo di path-loss e rumore al variare della frequenza per alcune distanze campione. In altre parole, il grafico rappresenta l'SNR calcolato con una potenza trasmittiva di $0 \text{ dB re } \mu\text{Pa}$.

La propagazione del suono, oltre a subire l'attenuazione e la distorsione del

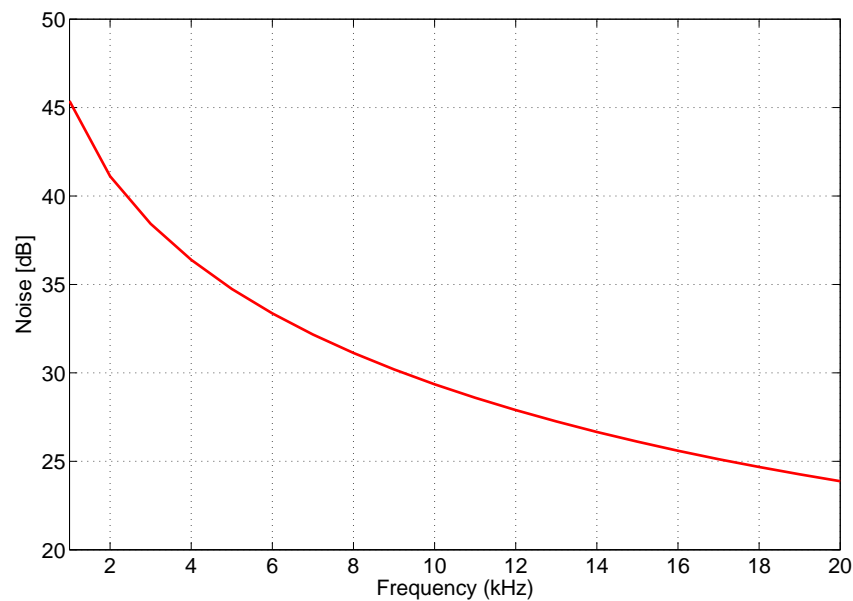


Figura 1.2: Contributo del rumore in dB al variare della frequenza in kHz.

canale sottomarino, dipende anche dalla temperatura e dalla pressione. La temperatura e la pressione, infatti, influenzano la variazione della densità dell'acqua e quindi della sua velocità. Il suono, come del resto i campi elettromagnetici, tende a propagarsi nelle zone ove la densità del mezzo di propagazione è minore, quindi ove la velocità è minore.

Quindi, legando la profondità e la temperatura del mare alla velocità di propagazione del suono, otteniamo nei primi metri di profondità una zona dipendente dalla temperatura, chiamata *termoclino*. Aumentando la profondità incontriamo una zona detta *termicamente stabile*, non soggetta al riscaldamento dovuto all'irraggiamento solare e, continuando ad aumentare la profondità, osserviamo un progressivo aumento della velocità del suono dovuta all'aumento della pressione. Queste considerazioni sulla velocità del suono in funzione della profondità del mare, vengono

1. UNA PANORAMICA SULLE COMUNICAZIONI ACUSTICHE SOTTOMARINE

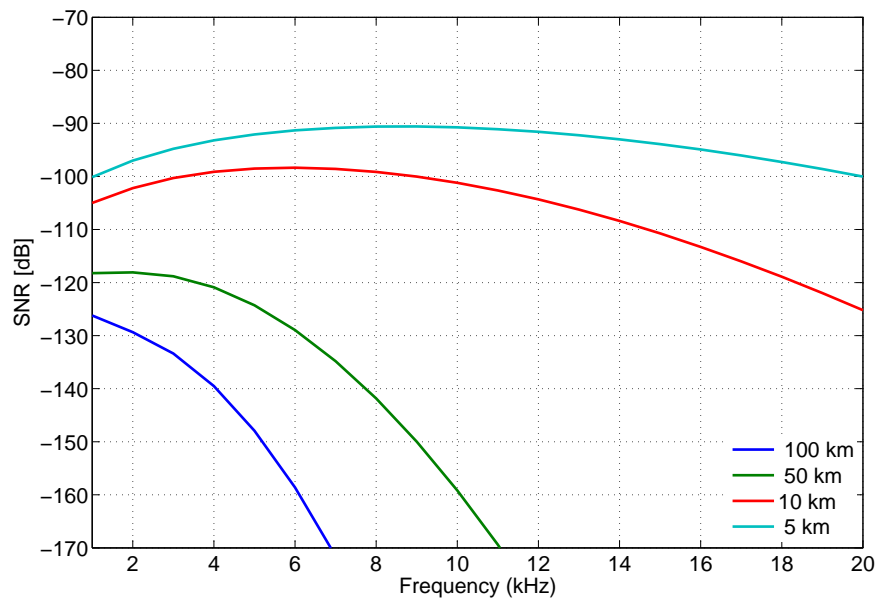


Figura 1.3: SNR con un potenza trasmisiva di $0 \text{ dB re } \mu\text{Pa}$ al variare della frequenza, per alcune distanze campione.

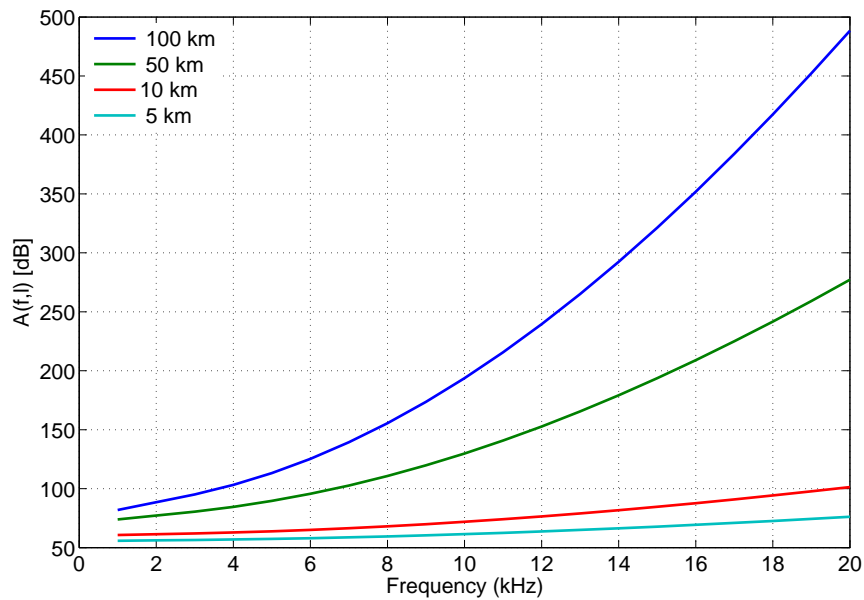


Figura 1.4: Andamento del fattore $A(l, f)$ al variare della frequenza per alcune distanze campione.

raccolte in una funzione chiamata *sound speed profile (SSP)*.

La Figura (1.5) mostra un SSP di esempio preso dall'oceano Atlantico nel mese

di Luglio. Il grafico presenta effettivamente l'andamento previsto teoricamente in questo paragrafo. La zona del grafico a velocità minore, dove effettivamente il suono tende a propagarsi, viene chiamato *canale SOFAR*. Bisogna tenere conto, quindi, specialmente per comunicazioni a distanza considerevole, che segnali acustici mandati al di sopra o al di sotto del *deep-sound axes* tendono a curvare verso il basso o verso l'alto, rispettivamente, con una curvatura tanto più accentuata quanto più accentuato è il gradiente di velocità tra la velocità del suono alla profondità di trasmissione e la velocità del suono nel *deep-sound axes*. Questa curvatura è dovuta al cambiamento graduale dell'indice di rifrazione n dell'acqua, a propria volta dovuta al cambiamento di densità del liquido stesso. Il suono risulta quindi confinato in questa zona, con un processo fisico che ricorda quello alla base del funzionamento delle fibre ottiche, cioè il confinamento per cambiamento graduale dell'indice di rifrazione. D'altro canto, potendo scegliere la profondità a cui posizionare i nodi (questa scelta dipende ovviamente dall'applicazione specifica), piazzarli nel *deep-sound axes* può essere conveniente per riuscire ad avere una propagazione meno attenuata e quindi riuscire a comunicare più lontano a parità di potenza trasmessa dei nodi.

La zona del *deep-sound axes* rappresenta, quindi, un "canale preferenziale" dove il suono si propaga con maggiore facilità. Tuttavia, questo canale non è costante e dipende dalla temperatura dell'acqua, soprattutto in superficie.

È evidente, quindi, come durante le ore della giornata le condizioni dell'SSP possano variare e durante le stagioni la presenza di questo "canale" possa essere più o meno accentuato.

In tutti gli SSP, sia invernali che estivi, al di sotto di una certa profondità la velocità del suono aumenta linearmente all'aumentare della profondità, tuttavia questa caratteristica diventa di poco interesse per i nostri scopi perchè al di sotto di una certa profondità i nodi non vengono generalmente posizionati a causa di una difficoltà oggettiva nella posa e nel successivo recupero. D'altra parte, i nodi più in superficie sono soggetti al cambiamento dell'SSP durante la giornata e durante le stagioni, proprio per i motivi di cui si parlava poco sopra.

In Figura (3.19) si può vedere il pathloss (la barra colorata rappresenta l'attenuazione in dB data dal canale) ottenuto tramite Bellhop e riferito all'SSP di Figura (1.5).

Osservando l'immagine, si può notare chiaramente l'effetto del canale SOFAR

1. UNA PANORAMICA SULLE COMUNICAZIONI ACUSTICHE SOTTOMARINE

a circa 1500 metri, in cui l'attenuazione è minore per una distanza maggiore.

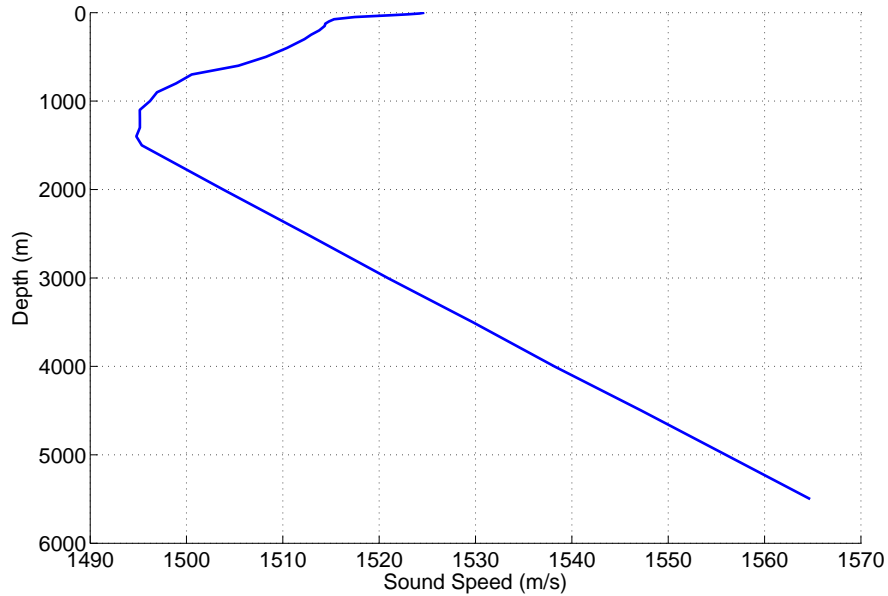


Figura 1.5: Sound Speed Profile di esempio dell'oceano atlantico in Luglio.

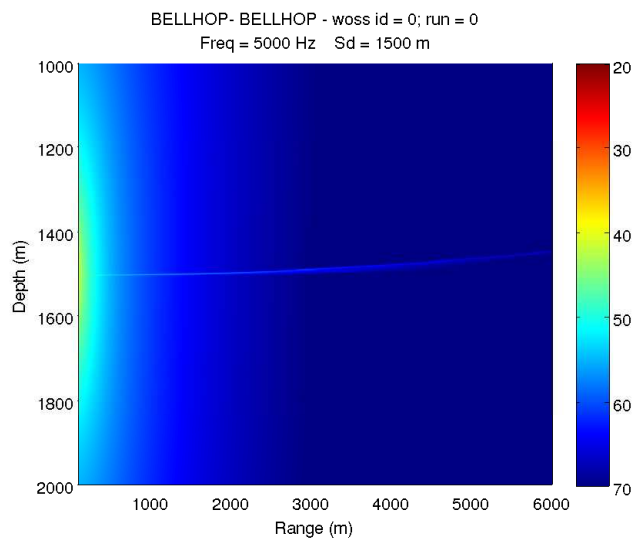


Figura 1.6: Path-loss ottenuto tramite Bellhop riferito all'SSP di Figura 1.5.

In Figura (1.7) e in Figura (1.8) sono riportati due esempi di SSP: uno tipicamente estivo (Agosto) e uno invernale (Gennaio), rispettivamente. I due grafici sono riferiti alla zona dell'Oceano Atlantico Settentrionale, precisamente alle coordinate

-
- **Latitudine:** 54° N
 - **Longitudine:** 39° O

Si può notare chiaramente come nell'SSP estivo sia presente il canale SOFAR, mentre non ci sia traccia del suddetto nel grafico invernale, dovuto al fatto che in inverno la superficie del mare non venga adeguatamente riscaldata dal sole.

In Figura (1.7) e in Figura (1.8) ci sono i rispettivi path-loss costruiti con l'ausilio di Bellhop. Possiamo notare come gli effetti del canale SOFAR siano evidenti nel periodo estivo, in particolare ci siano delle zone che curvano verso il basso (cioè proprio verso la profondità del deep-sound axes) il cui colore è un blu meno scuro, sintomatico del fatto che in quelle zone l'attenuazione è minore.

Nel periodo invernale, invece, questi effetti sono praticamente assenti.

Lo scopo di questa tesi è quello di investigare quali protocolli MAC presentano delle performance migliori in tutti questi ambienti molto diversi tra loro, con condizioni di propagazione molto diverse tra di loro.

Il fine ultimo è quello di ricercare un protocollo che abbia delle performances buone sotto il punto di vista del throughput o del Packet Error Rate o di qualsiasi altra metrica di rete in tutti gli ambienti, trovando il candidato ideale per diventare uno standard di uso comune nelle comunicazioni acustiche sottomarine. Successivamente si potrà cercare di prendere i lati positivi da ciascun protocollo creandone un "ibrido" che abbia delle performances ottime in qualsiasi ambiente e in qualsiasi condizione di propagazione, a seconda dell'applicazione richiesta.

1. UNA PANORAMICA SULLE COMUNICAZIONI ACUSTICHE SOTTOMARINE

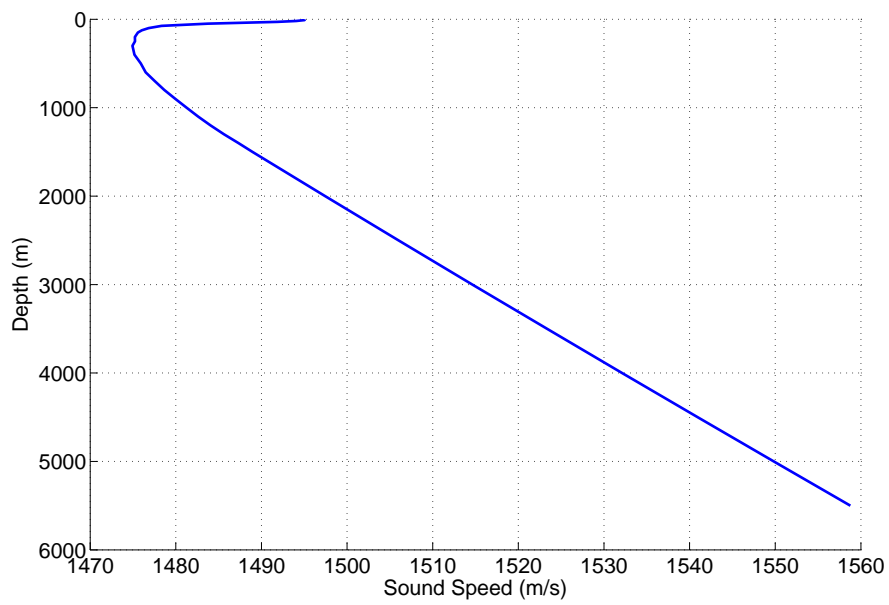


Figura 1.7: Sound-speed profile estivo.

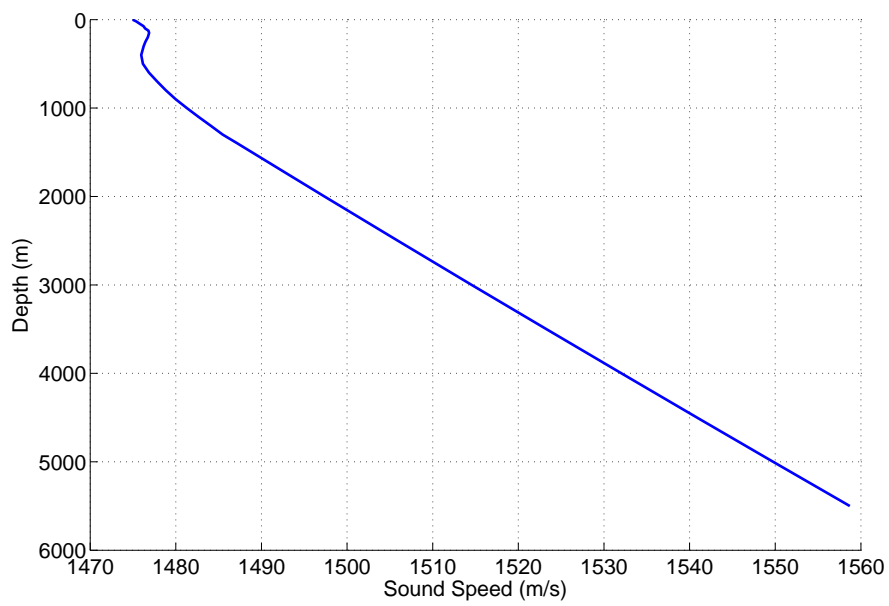


Figura 1.8: Sound speed profile invernale.

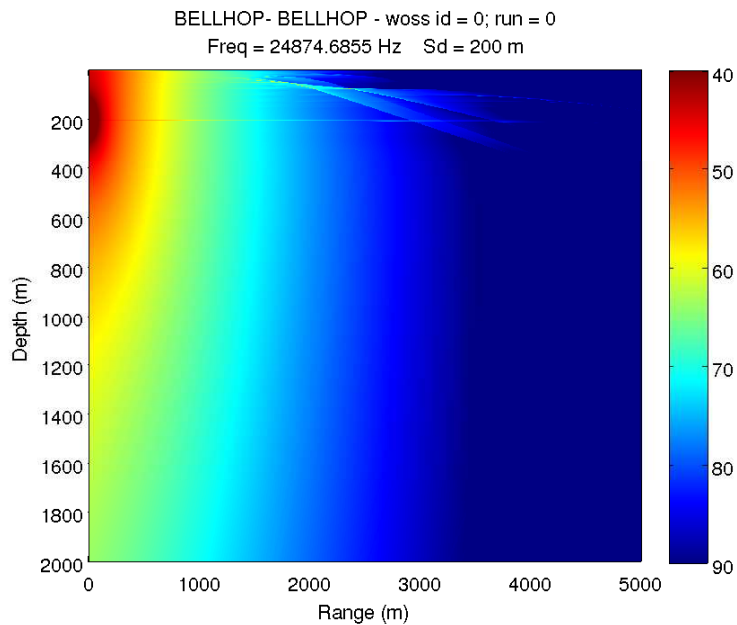


Figura 1.9: Path-loss legato all'SSP di Figura (1.7).

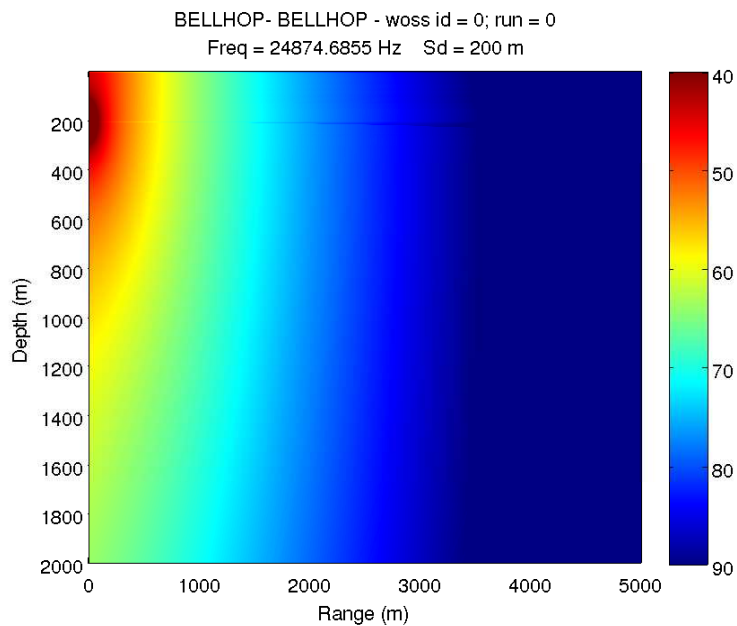


Figura 1.10: Path-loss legato all'SSP di Figura (1.8).

1.2 Caratteristiche di livello fisico

La modulazioni utilizzate nelle comunicazioni acustiche sottomarine sono di cardinalità solitamente bassa. Le modulazioni con cardinalità maggiore offrono un bit-rate maggiore a parità di baud-rate, ma la loro costellazione è meno immune

1. UNA PANORAMICA SULLE COMUNICAZIONI ACUSTICHE SOTTOMARINE

al rumore. Nel seguito di questa tesi, per le applicazioni di interesse (valutazione di prestazione di protocolli di accesso al mezzo) , utilizzeremo sempre una modulazione *BPSK*.

Questa modulazione è binaria, la più semplice della classe delle modulazioni *PSK* (Phase Shift Keying) ma la più immune al rumore.

In alcuni modem vengono inoltre utilizzate modulazioni di tipo *FSK* (Frequency Shift Keying) sempre con una cardinalità molto bassa.

A quest'ultima modulazione viene spesso affiancata una tecnica di *Frequency Hopping*, ottenendo uno schema detto FH-FSK. La tecnica di Frequency Hopping permette di avere una comunicazione efficiente in un ambiente con canali fortemente multipath come quelli underwater, specialmente se l'ambiente è di tipo "shallow-water" in cui le riflessioni sono molteplici e dovute al fatto che il fondale e la superficie sono relativamente vicini alla zona di trasmissione del suono [15]. Per quanto riguarda i bit-rate utilizzati, questi sono relativamente bassi se paragonati ad applicazioni terrestri.

Per i nostri scopi sono state utilizzate velocità dagli 800 bps ai 4800 bps. In [15], il modem presentato ha un bit-rate di default di 80 bps con la modulazione FH-FSK; mentre i bit-rates possibili tramite la modulazione PSK vanno dai 300 ai 5000 bps.

L'equalizzazione, nel modem preso in esame [15], è demandata a un DSP apposito che implementa un equalizzatore DFE (Decision Feedback Equalizer). Questo DSP ha dei consumi considerevoli per eseguire le operazioni di equalizzazione richieste.

Tuttavia questo DSP viene disattivato durante la fase di *sleep* ed è in grado di attivarsi quando un segnale PSK viene ricevuto dal modem acustico.

Per quanto riguarda la frequenza di lavoro, in generale, si va da pochi kHz ad alcune decine di kHz. L'utilizzo di frequenze più alte è limitato principalmente dal fatto che alzando la frequenza aumenta in maniera molto veloce l'attenuazione. Come si può vedere dalla Figura (1.1), per rimanere al di sotto di un valore di attenuazione di 50 dB/km, si è costretti a rimanere al di sotto dei 50 kHz. D'altro canto, come mostra il grafico (1.2), a frequenze molto basse abbiamo un contributo non indifferente di rumore. Dobbiamo quindi trovare un buon compromesso tra avere una bassa attenuazione e un basso contributo di rumore. Nelle nostre applicazioni, comunque, utilizzeremo delle frequenze tra i 5 kHz e i

25 kHz. Anche la banda nei canali underwater è molto ristretta. Come già spiegato precedentemente, l'attenuazione del canale acustico sottomarino dipende anche dalla frequenza oltre che dalla distanza. Questo fatto pone un limite alla larghezza di banda dei canali, partendo dal presupposto di porre un limite massimo di potenza trasmissibile. Come conseguenza immediata di questo fatto, si ha che un link più corto ha una larghezza di banda maggiore rispetto a un link più lungo. Quindi, una comunicazione lunga 100 km può essere fatta tramite un canale con larghezza di banda di circa 1 kHz, oppure utilizzando 10 hops con link di 1 km, quindi 10 canali di 10 kHz ciascuno, aumentando così il throughput, a spese di un più complicato sistema di relay e di protocollo di routing.

I valori tipici di banda utilizzata si aggirano comunque sempre su poche decine di kHz, tipicamente 5 kHz.

Le potenze trasmissive in gioco, invece, dipendono fortemente dall'applicazione e dalla distanza che si decide di coprire. Si parla, tuttavia, di alcune centinaia di dB *re* μPa (unità di misura della pressione acustica) per coprire link della lunghezza di qualche km.

Si tenga conto comunque che il modem WHOI [15] richiede il consumo di 50 W di potenza durante una trasmissione a 190 dB *re* μPa . Il consumo di potenza è un dettaglio da non trascurare perchè la durata della batteria è un aspetto critico in applicazioni sottomarine, dove la sostituzione della batteria può richiedere tempo e risorse. Per questo motivo, si cerca di costruire modem che riescono a passare la maggior parte del tempo in uno stato *quiet*, per poi passare in uno stato di *wake-up* alla ricezione di un segnale valido.

A questo proposito anche i protocolli sono progettati con un occhio di riguardo all'efficienza energetica e al risparmio di potenza.

Capitolo 2

I protocolli di accesso al mezzo per le comunicazioni underwater

In questa sezione sono presentati i protocolli di accesso al mezzo che verranno analizzati nella seconda parte della tesi.

Questi protocolli sono sostanzialmente adattamenti di protocolli già utilizzati in ambito terrestre, sia in reti via cavo che wireless.

Questi protocolli si possono distinguere per il livello di coordinamento richiesto. Si va da una mancanza totale di coordinazione tra i nodi (CSMA-ALOHA), a una coordinazione leggera e guidata dal trasmettitore (TONE-LOHI), a una coordinazione pesante (DACAP). In altre parole, i protocolli con una grande coordinazione, cercano di tenere sotto controllo le attività di tutti i nodi e cercano di fare in modo che essi (o almeno i più vicini nella rete) sappiano cosa sta succedendo e chi necessita di trasmettere in un determinato momento. Ovviamente, per far questo, è necessaria la trasmissione di molti pacchetti di controllo. In caso di basso traffico, questa grande segnalazione, porta a una certa inefficienza della rete dal punto di vista del throughput utile.

Da una prima sommaria analisi, appare intuitivamente sensato che i protocolli con un certo grado di coordinazione abbiano delle prestazioni migliori rispetto ai protocolli che trasmettono i pacchetti lasciandoli sostanzialmente al caso (come fa d'altra parte il protocollo ALOHA puro). Vedremo, nel prossimo capitolo, che elevata coordinazione non è necessariamente sinonimo di elevate prestazioni, a causa delle condizioni molto diverse nelle comunicazioni acustiche sottomarine, come gli elevate latenze dovute al tempo di trasmissione del segnale.

Il protocollo USB-AIMD [5], implementa invece una politica di ARQ (Automatic Repeat reQuest) sul protocollo CSMA-ALOHA. In questo caso, quindi, il pacchetto verrà trasmesso seguendo le regole imposte dal protocollo CSMA-ALOHA, ma il trasmettitore si aspetterà un riscontro da parte del ricevitore (ACK). In mancanza di questo riscontro, verranno appunto messe in atto politiche di ritrasmissione.

2.1 CSMA-ALOHA

Il protocollo ALOHA è un protocollo ad accesso casuale.

Appena i nodi hanno un pacchetto da trasmettere, lo trasmettono senza alcuna coordinazione. In questo caso, quindi, le collisioni avvengono quando due nodi tentano di accedere al canale allo stesso tempo [2].

In caso di collisione, il trasmettitore aspetta un tempo random tra 0 e il doppio del massimo tempo di propagazione della rete prima di ritentare. A ogni fallimento di trasmissione, il range in cui viene scelto il tempo di backoff viene raddoppiato per un massimo di cinque tentativi consecutivi di trasmissione. A ogni trasmissione con successo, il range di tempo di backoff viene resettato al minimo.

Il protocollo ALOHA, così come è costruito, pur avendo dei tempi morti molto bassi e poco overhead dovuto alla sostanziale assenza di segnalazione di controllo, è soggetto a molte collisioni se la rete è abbastanza densa e il traffico è molto alto. L'efficienza del protocollo stesso si ricava facilmente.

Normalizziamo a 1 il tempo necessario alla trasmissione di un pacchetto, $T_D = 1$. Supponiamo ora che un nodo cominci la trasmissione di un pacchetto al tempo t_0 . La trasmissione del pacchetto può considerarsi completa con successo se nessun altro nodo nella rete trasmette nell'intervallo di tempo $[t_0 - 1, t_0]$, in quanto si sovrapporrebbe con la trasmissione del pacchetto del nodo di nostro interesse. Parimenti, nessun altro nodo può trasmettere mentre il nodo in esame sta trasmettendo il suo pacchetto, in altre parole nessun altro nodo può trasmettere nell'intervallo $[t_0, t_0 + 1]$, altrimenti i due pacchetti si sovrapporrebbero nella parte finale.

Definita p la probabilità che un nodo stia trasmettendo, le probabilità che i due eventi appena nominati sopra non si verifichino sono uguali e si calcolano come

$$P_1 = P_2 = (1 - p)^{N-1} \tag{2.1}$$

2. I PROTOCOLLI DI ACCESSO AL MEZZO PER LE COMUNICAZIONI UNDERWATER

dove N è il numero di nodi nella rete. La probabilità che entrambi gli eventi non si verificano contemporaneamente è il prodotto delle due probabilità ottenendo quindi

$$P = P_1 \cdot P_2 = (1 - p)^{2(N-1)} \quad (2.2)$$

Per ottenere l'efficienza massima per un numero elevato di nodi si esegue il limite della probabilità (2.2), ottenendo

$$\lim_{N \rightarrow \infty} P = \frac{1}{2e} \quad (2.3)$$

che rappresenta una efficienza abbastanza bassa [16]. Per i nostri scopi, il protocollo è stato arricchito di una politica standard di risoluzione delle contese: prima di trasmettere, il protocollo ascolta istantaneamente il canale e se lo trova libero trasmette, altrimenti postpone la trasmissione finché capta energia dal canale.

Il protocollo che andremo a utilizzare, quindi, non è un ALOHA puro ma implementa un semplice sensing di canale. Questo sensing, pur essendo molto breve, permette di rendere più robusto il protocollo e permette di evitare una parte delle collisioni, che invece il protocollo ALOHA puro non riuscirebbe a evitare non controllando assolutamente lo stato del canale.

L'associazione di questi due protocolli risulta vincente in ambito underwater, dove i lunghi tempi di propagazione permettono ai nodi di trasmettere completamente il pacchetto senza altre sovrapposizioni se il traffico di dati nella rete non è elevato.

Si ricorda, inoltre, che il protocollo *Slotted-ALOHA*, in cui gli slot dove è permessa la trasmissione devono essere dimensionati tenendo conto del tempo di trasmissione di un pacchetto e del tempo massimo di propagazione del segnale, non necessariamente ha delle prestazioni migliori rispetto all'ALOHA puro come invece succede nelle reti terrestri. Il vantaggio della riduzione del tempo di vulnerabilità del protocollo è annullato dal fatto che gli slot risultano essere molto lunghi, poiché i tempi di propagazione in ambiente sottomarino possono essere considerevolmente lunghi. Essendo gli slot molto lunghi, quindi, aumentano a dismisura i tempi morti del protocollo stesso, che come effetto diretto porta a una diminuzione del throughput nella rete.

2.2 DACAP

Il protocollo DACAP (Distance Aware Collision Avoidance Protocol) [3] è un protocollo non sincronizzato che si basa sullo scambio di messaggi RTS/CTS per instaurare una connessione e trasmettere i dati. Questo scambio di messaggi è effettuato per evitare le interferenze distruttive di eventuali messaggi spediti da nodi limitrofi ai due nodi impegnati nella comunicazione. Il protocollo è specificato come segue.

Appena un nodo ha dati pronti da trasmettere, manda un messaggio RTS (*Request to Send*) (solitamente di pochi byte) in unicast al destinatario della comunicazione. Il ricevitore, appena ricevuto l'RTS risponde immediatamente con un CTS (*Clear to Send*) e si mette in ascolto del canale per ricevere il pacchetto dati. Se in questo periodo di attesa del payload, sente un messaggio RTS proveniente da un altro nodo vicinante, avvisa il trasmettitore tramite un piccolo pacchetto WRN (*Warning*). È essenziale che il pacchetto WRN sia di pochissimi byte, in modo da essere trasmesso velocemente, cioè prima che il trasmettitore inizi la trasmissione del pacchetto e quindi diventi “sordo” in ricezione (si ricorda che i modem acustici in uso sono half-duplex).

Il trasmettitore, appena ricevuto il messaggio CTS, non si mette a trasmettere subito il payload ma attende un certo tempo. Se in questo periodo di tempo, riceve un WRN dal suo partner oppure sente un CTS da un altro nodo confinante ed estraneo alla comunicazione, annulla la comunicazione, evitando l'interferenza che ne sarebbe scaturita.

Il protocollo sfrutta la notevole potenza ricevuta se il link è breve e parte dal presupposto che un pacchetto può essere comunque decodificabile anche se è sovrapposto a un'altra comunicazione, a patto che l'interferente che ha trasmesso il pacchetto sia sufficientemente lontano, in modo che il segnale desiderato abbia un SINR (*Signal to Interference plus Noise Ratio*) abbastanza alto. Infatti, se l'interferente è sufficientemente lontano, il segnale interferente avrà già subito una attenuazione del canale tale da avere una potenza bassa a sufficienza per non coprire il segnale desiderato e permettere che questo sia ancora decodificabile con una qualità che a priori verrà decisa.

Il tempo di attesa del trasmettitore, può essere configurabile dall'utente, che in pratica decide quanto lontano vuole che si propaghi il suo RTS e, come diretta conseguenza, decide quanti nodi devono venire a conoscenza della presenza di

2. I PROTOCOLLI DI ACCESSO AL MEZZO PER LE COMUNICAZIONI UNDERWATER

una comunicazione. Quindi, un nodo aspetterà un tempo sufficiente per fare in modo che il pacchetto si propaghi a una certa distanza oltre la distanza del link. Passato questo tempo, che chiamiamo T_w (tempo di *warning*), il nodo trasmette il suo payload. I nodi più distanti non avranno fatto in tempo a ricevere l'RTS, ma vengono ritenuti talmente distanti da non poter nuocere alla trasmissione in oggetto.

Definita U come la distanza tra trasmettitore e ricevitore (il RTT può essere calcolato attraverso lo scambio CTS/RTS) e definita ΔD la distanza aggiuntiva in cui l'utente decide di far propagare il messaggio RTS (in pratica oltre la distanza $U + \Delta D$ i nodi vengono considerati talmente distanti da non poter interferire distruttivamente con la comunicazione), il protocollo calcola T_w come

$$T_w = \begin{cases} t_{min} - \frac{2U}{c} & \text{se } \frac{U}{c} < t_1 \\ \frac{2(U+\Delta D)}{c} - t_{min} & \text{se } \frac{U}{c} \in (t_1, t_2) \\ \frac{2D}{c} + t_{data} & \text{se } \frac{U}{c} > t_2 \end{cases} \quad (2.4)$$

dove

$$t_1 = \frac{t_{min} - \min(D/c, t_{data})}{2} \quad (2.5)$$

$$t_2 = \frac{t_{data} + t_{min}}{2} \quad (2.6)$$

con la restrizione ulteriore che sia verificata la disuguaglianza $T_w > 2D/c$.

Nelle formule soprastanti U , come già anticipato, è la distanza tra trasmettitore e ricevitore calcolata tramite lo scambio RTS/CTS, c è la velocità del suono settata per default a 1500 m/sec . t_{data} è invece il tempo di trasmissione di un pacchetto. Il parametro t_{min} è il tempo minimo per eseguire un handshake. Definito preliminarmente il massimo range di trasmissione, T è il tempo di propagazione del segnale per il suddetto range.

t_{min} dovrebbe essere assegnato a $2T$ (tempo di propagazione di RTS e CTS alla massima distanza). Nel progetto del protocollo, invece, si è partiti dal presupposto che non tutti i nodi saranno alla massima distanza di propagazione, ma molti, invece, saranno più vicini. In questo senso, quindi, impostare il tempo di handshake a $2T$ sarebbe uno spreco di tempo, perchè nella maggior parte dei casi ci vorrebbe molto meno tempo per eseguire un handshake. Il parametro t_{min} permette di fare proprio questo. Se impostato in maniera che $t_{min} < 2T$, permette di avere degli handshake più veloci di $2T$ con un throughput che aumenta. Quanto

minore deve essere questo parametro, ovviamente, dipende dalla struttura della rete. Per dimensionarlo correttamente bisognerebbe conoscere perfettamente la struttura della rete, ma non sempre è possibile.

Si esegue quindi una statistica che permette, se possibile, di conoscere la distanza media dei nodi in maniera di impostare t_{min} in maniera che sia il tempo minimo per eseguire un handshake per la maggior parte dei nodi, fermo restando il parametro $2T$ che rappresenta il massimo tempo possibile per l'instaurazione di una connessione.

Da una primo sommario confronto, come già preannunciato nell'introduzione, il protocollo DACAP ha molti più tempi "morti" rispetto al CSMA-ALOHA. Il rovescio della medaglia sta nel fatto che, in teoria, la comunicazione dovrebbe essere esente da interferenza perchè tutti i nodi sono a conoscenza della presenza di una comunicazione.

Ovviamente in sede di progetto è importante dimensionare bene il parametro D . Impostare questo parametro troppo grande significa avere dei tempi di attesa troppo grandi; la comunicazione sarà più sicura ma il throughput sarà comunque molto basso a causa dei tempi esigui dedicati alla trasmissione di dati utili. D'altro canto, impostare D troppo piccolo significa avere dei tempi di attesa minori ma una probabilità più alta che una interferenza vada a distruggere la nostra comunicazione.

Il calcolo del valore ottimo D può essere fatto tramite considerazioni sul SINR, vediamo meglio queste considerazioni tramite un esempio.

In Figura (2.1) vediamo l'andamento del SINR in funzione del parametro D (in figura indicato come ΔD) per varie distanze di trasmissione U con i seguenti parametri.

- Modulazione BPSK
- Frequenza di lavoro $f = 25 \text{ kHz}$,
- Potenza di trasmissione del nodo in esame e dell'interferente pari a $P = 150 \text{ dB re } \mu\text{Pa}$,
- Fattore di spreading $k = 1.5$,
- Shipping noise factor $s = 0.5$,
- Assenza di vento $w = 0$

2. I PROTOCOLLI DI ACCESSO AL MEZZO PER LE COMUNICAZIONI UNDERWATER

La linea orizzontale tratteggiata, invece, rappresenta una ipotetica sensibilità del ricevitore posta a 5 dB.

Nel grafico si vede l'andamento del SINR per una certa distanza di trasmissione, con il nodo interferente posto a una distanza pari a quella del ricevitore più la variabile D da settare nel protocollo. Dal grafico risulta chiaro che l'interferente comincia a essere ininfluenza per la trasmissione dopo i 3500 metri.

In Figura (2.2) si vede invece l'andamento del PER in funzione del parametro D , che per fare le considerazioni precedenti, risulta più chiaro. Ovviamente il grafico di Figura (2.1) e il grafico di Figura (2.2) sono legati tramite la formula per il calcolo del PER attraverso il SINR per una modulazione BPSK

$$PER = 1 - (1 - 0.5Q(\sqrt{SINR})^L) \quad (2.7)$$

dove SINR è espresso in scala lineare e L è la dimensione del pacchetto in bit (nel nostro caso 1000 bit).

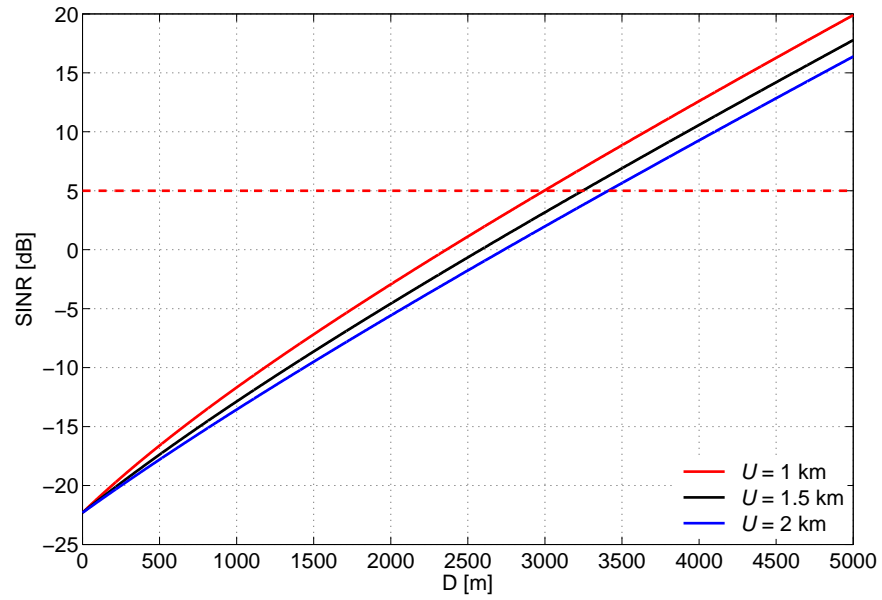


Figura 2.1: Andamento del SINR in funzione del parametro D per varie distanze di trasmissione.

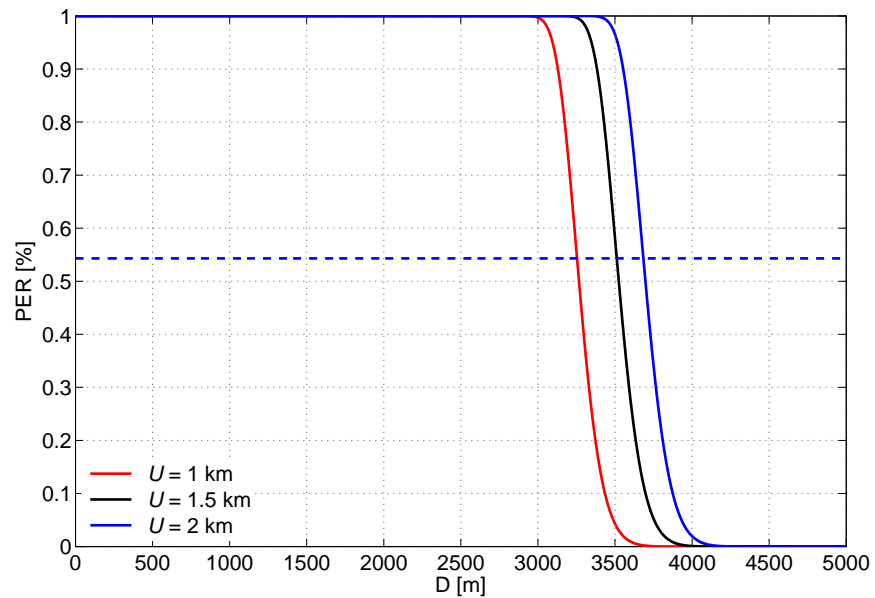


Figura 2.2: Andamento del PER in funzione del parametro D per varie distanze di trasmissione.

2.3 TONE-LOHI

Il protocollo TONE-LOHI [17] è un protocollo *reservation-based* basato su un leggero coordinamento per la contesa del canale guidato dal trasmettitore. Ogni “frame” T-Lohi consiste in una fase preliminare di contesa chiamata RP (*Reservation Period*) seguita dall’effettiva trasmissione di dati. Ogni RP è formato da uno o più CR (*Contention Round*). In ogni CR un nodo manda un breve tono e poi imposta l’hardware in ascolto di eventuali altri toni da parte dei contendenti fino alla fine del periodo di CR. Se alla fine di questo periodo non ha percepito altri toni, si dichiara vincitore della contesa e comincia la trasmissione dei dati. Se più di un nodo concorre all’acquisizione del canale, ogni nodo sentirà il tono degli altri e andrà in backoff per un periodo random proporzionale al numero di toni contati. In questa maniera il numero di CR aumenterà finchè non verrà dichiarato il vincitore.

In Figura (2.3) possiamo vedere un esempio di funzionamento del protocollo.

La particolarità del canale underwater di avere dei tempi di propagazione molto elevati permette ai nodi non solo di capire che ci sono altri contendenti nel canale (*CTD-Contender Detection*) ma anche di contare il numero di contendenti stessi (*CTC - Contender Counting*) se il tono è breve a sufficienza e, come spiegato sopra, il backoff random dei nodi è calcolato proprio in base al CTC.

Alcuni modem sono dotati di hardware che permette di avere un sensore a basso consumo di energia che ascolta il canale cercando la presenza di toni, svegliando la parte radio solo in presenza di pacchetti dati. Questa politica permette di risparmiare molta energia nei nodi stessi, con una grande salvaguardia della batteria stessa.

Se il nodo non è dotato di questa caratteristica, i toni possono essere sostituiti da messaggi RTS di pochissimi byte, piccoli a sufficienza per poter essere contati distintamente dal ricevitore.

Uno dei parametri fondamentali da settare per questo tipo di protocollo è la durata del *Contention Round*. A seconda della durata del CR, possiamo distinguere tre versioni differenti dello stesso protocollo.

2.3.1 Synchronized TONE-LOHI (ST-Lohi)

In questa versione del protocollo tutti i nodi sono sincronizzati nel tempo. La durata del CR è $CR_{ST} = \tau_{max} + T_{tone}$ dove τ_{max} è il tempo massimo di propagazione unidirezionale e T_{tone} è il tempo minimo di durata del tono per essere ricevuto dal nodo (parametro che dipende dal tipo di hardware). Considerato che tutti i nodi sono sincronizzati nel tempo, ogni tono viene mandato all'inizio del CR e c'è la sicurezza che il tono raggiungerà tutti i nodi prima della fine del periodo di CR.

Questa versione del protocollo dovrebbe garantire trasmissioni prive di collisioni nella maggior parte dei casi. D'altra parte, però ci sarà una spesa dal punto di vista della messaggistica per mantenere sincronizzati i nodi tra di loro.

2.3.2 Conservative Unsynchronized TONE-LOHI (cUT-Lohi)

In questo caso, i nodi possono cominciare il CR in qualsiasi momento se sanno che il canale non è impegnato. Per evitare le collisioni, il CR deve durare $CR_{cUT} = 2\tau_{max} + 2T_{tone}$. Non essendo i nodi sincronizzati, si deve duplicare il tempo di attesa. Rispetto al modello precedente, non dobbiamo complicare il protocollo per ottenere una sincronizzazione dei nodi stessi, ma dobbiamo, d'altra parte, duplicare i tempi di contesa, dedicando meno tempo alla trasmissione, a spese del throughput.

2.3.3 Aggressive Unsynchronized TONE-LOHI (aUT-Lohi)

In quest'ultima versione, i nodi sono ancora non sincronizzati, quindi, come prima, possono cominciare il CR appena sentono il canale libero e hanno dati pronti per la trasmissione. In questa versione, però, il contention round dura $CR_{aUT} = \tau_{max} + T_{tone}$ come nel caso ST-Lohi.

Questa versione del protocollo risponde proprio all'esigenza di accorciare i tempi di attesa che vanno a impattare sul throughput, nonostante la presenza di collisioni sia più probabile visto che si ascolta il canale per metà del tempo prima di trasmettere i dati.

Contrariamente a quanto si possa pensare, alcuni risultati in [18] e [17] dimostra-

2. I PROTOCOLLI DI ACCESSO AL MEZZO PER LE COMUNICAZIONI UNDERWATER

no che questa versione ha delle prestazioni migliori in termini di throughput.

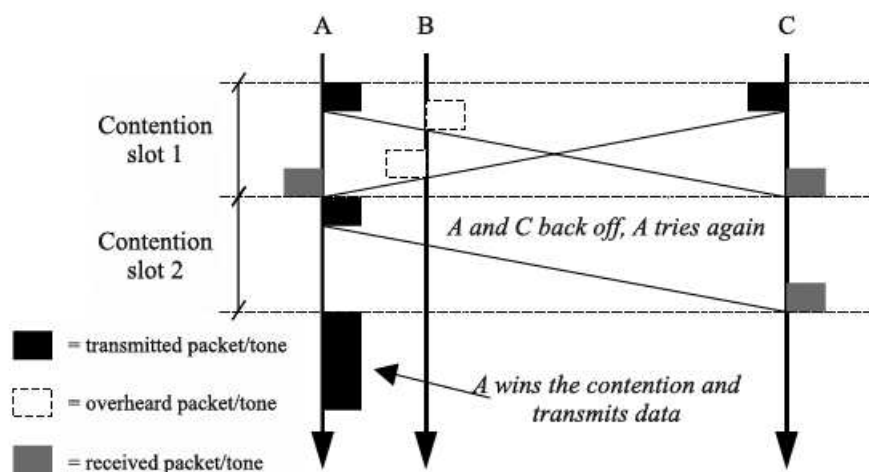


Figura 2.3: Schema di funzionamento del protocollo TONE-LOHI [18].

2.4 USR-AIMD

Il protocollo USR-AIMD [5] implementa una politica di ARQ (*Automatic Repeat Query*) al di sopra di un protocollo ad accesso casuale già esistente, nel nostro caso al di sopra del protocollo CSMA-ALOHA già descritto in questo capitolo.

Esistono sostanzialmente 3 implementazioni di un protocollo ARQ:

- **Stop and Wait (S&W)**: In questo caso il mittente invia un pacchetto e si ferma per un certo tempo deciso da un time-out in attesa di un ACK. Se alla fine di tale time-out l'ACK non è arrivato, re-invia il pacchetto
- **Go Back N (GBN)**: Nel GBN, il trasmettitore può trasmettere un numero di pacchetti fino ad N. Il ricevitore trasmette un ACK per ogni pacchetto informando il trasmettitore del numero dell'ultimo pacchetto ricevuto correttamente. Il trasmettitore, quindi, shifta la finestra a seconda del primo pacchetto non ricevuto correttamente. Se, per esempio, alla fine della trasmissione di tutti i pacchetti della finestra non è arrivato nemmeno un ACK, il trasmettitore ricomincia la trasmissione dall'inizio della finestra.

-
- **Selective Repeat (SR):** In quest'ultimo caso il ricevitore può mandare ACK in maniera disordinata al trasmettitore. Ovviamente al ricevitore bisogna re-sincronizzare il buffer per ordinare i pacchetti.

Se il GBN è estremamente inefficiente perchè anche un solo errore può provocare l'intera ri-trasmissione anche di pacchetti che erano arrivati correttamente, S&W, invece, è molto utilizzato perchè molto semplice da implementare e soprattutto si adatta bene alla caratteristica del canale underwater di essere half-duplex. Tuttavia, a causa degli elevati tempi di propagazione in ambito sottomarino, una politica S&W può diventare estremamente inefficiente perchè maggior parte del tempo viene speso attendendo l'arrivo dell'ACK, mentre, in un solo RTT possono essere trasmessi più pacchetti. Il protocollo SR, quindi, diventa più performante rispetto al S&W.

Utilizzando l'SR, però, bisogna implementare un canale full-duplex utilizzando un duplexing nel tempo (TDD) (l'ipotesi FDD è stata scartata a causa della grande selettività in frequenza del canale). Il protocollo USR (Underwater Selective Repeat) parte in modalità S&W: l'invio di un pacchetto e la ricezione del conseguente ACK permettono al trasmettitore di misurare il round-trip time; se l'ACK non viene ricevuto all'interno di un time-out, il protocollo re-invia il pacchetto utilizzando ancora il S&W, altrimenti si passa alla fase successiva.

Ponendo τ il tempo di propagazione tra il trasmettitore e il ricevitore, T_D il tempo di trasmissione di un pacchetto dati e T_A il tempo di trasmissione dell'ACK, se $\tau > T_D + T_A$ significa che si può un altro pacchetto nello stesso RTT e quindi il protocollo passa a una politica SR.

Per evitare di ricevere un ACK durante la trasmissione di un pacchetto, il trasmettitore aspetta un tempo $W = \tau + \Delta$ in cui Δ è un tempo di guardia che tiene conto di eventuali cambi di SSP.

Se, invece, il nodo ha in coda un solo pacchetto da trasmettere o se si ha $\tau < T_D + T_A$ il protocollo ritorna in modalità S&W; in particolare, il secondo caso (partendo dal presupposto che i dati siano di lunghezza fissa) si verifica solo per il cambiamento del parametro τ conseguente a un cambio sostanziale della velocità del suono, cioè del Sound-Speed Profile.

Questa versione del protocollo, inoltre, implementa una sorta di *Congestion Window* come nel caso del protocollo TCP chiamata AIMD (*Additive Increase-Multiple Decrease*) in cui questa finestra di trasmissione dei pacchetti viene dimez-

2. I PROTOCOLLI DI ACCESSO AL MEZZO PER LE COMUNICAZIONI UNDERWATER

zata a ogni ACK mancato e incrementata a ogni pacchetto trasmesso con successo. Come già detto in precedenza, il protocollo USR-AIMD può essere utilizzato congiuntamente a vari protocolli, ma nel nostro caso è stato utilizzato il protocollo CSMA-ALOHA, quindi un protocollo ad accesso casuale con un sensing iniziale del canale. Proprio il sensing del canale discrimina due diverse versioni del protocollo. In USR-v1, il trasmettitore effettua il sensing del canale solo prima di trasmettere il primo pacchetto in modalità S&W e non la effettua più fino alla perdita di un pacchetto o finché il nodo ha pacchetti da mandare allo stesso ricevitore.

In USR-v2, invece, il sensing viene effettuato prima di trasmettere tutti i pacchetti (subito dopo che l'ACK di quel pacchetto è stato ricevuto o subito che il time-out di attesa dell'ACK è scaduto) USR-v2 risulta molto più robusto alle interferenze e inoltre può essere utilizzato nel caso di nodi mobili, in cui il RTT cambia continuamente a causa della mobilità.

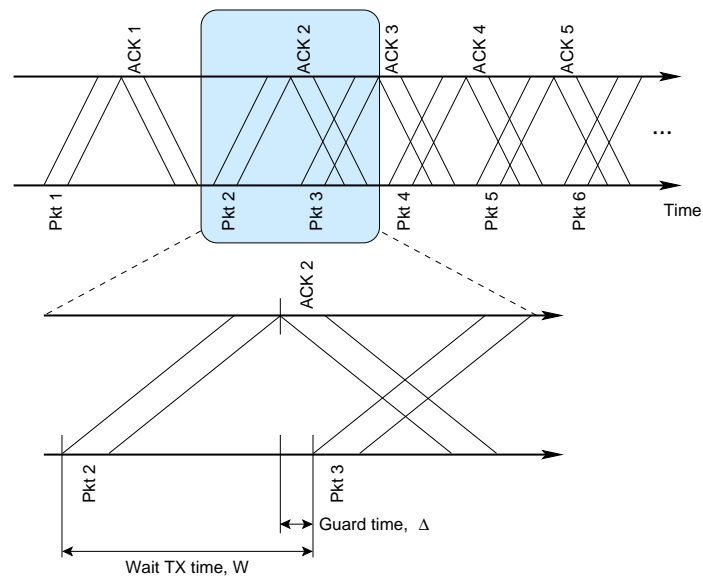


Figura 2.4: Esempio di funzionamento del protocollo USR-AIMD [5].

Capitolo 3

Simulazioni e risultati

I due scenari prescelti per le simulazioni cercano di coprire una vasta gamma di condizioni che si possono incontrare nelle comunicazioni underwater.

A differenziare le due zone prescelte, oltre all'SSP diverso, è la profondità del mare in cui vanno posizionati i nodi; per distinguere i due casi utilizzeremo i termini inglesi di uso comune in letteratura, cioè *shallow-water* per indicare un mare poco profondo, in cui i nodi occupano tutta la zona che va dalla superficie fino al fondale e *deep-water* per indicare i casi (come nell'oceano) dove la profondità del mare supera qualche migliaio di metri.

Le propagazioni dei segnali nei due casi sono molto diversi. In particolare, nel caso *deep water* è presumibile che non ci siano molte riflessioni sul fondale o sulla superficie, se i nodi sono piazzati a una profondità media e sono fluttuanti; la propagazione, quindi, può essere considerata sferica e il canale non è fortemente multipath.

Nel caso *shallow-water*, il fondale e la superficie dell'acqua sono relativamente vicini ai nodi. La trasmissione, di conseguenza, subisce molte riflessioni e il canale si presenta fortemente multipath. La riflessione del suono dipende dalla tipologia di fondale (batimetria, materiale presente nel fondale), queste informazioni verranno ricavate dai database [10] e [11] ed elaborate da WOSS.

Per quanto riguarda lo scenario deep-water, abbiamo scelto una zona nel nord dell'Oceano Atlantico, precisamente alle coordinate (**37.5 °N, 38.5 °O**) (Figura 3.1). In Figura (3.2) e (3.3) riportano gli SSP estivo e invernale rispettivamente (Luglio e Gennaio).

Come si evince dai grafici, durante tutto l'arco dell'anno sono presenti dei canali

3. SIMULAZIONI E RISULTATI

SOFAR (Sound Fixing And Ranging), cioè dove il suono ha un minimo della velocità.

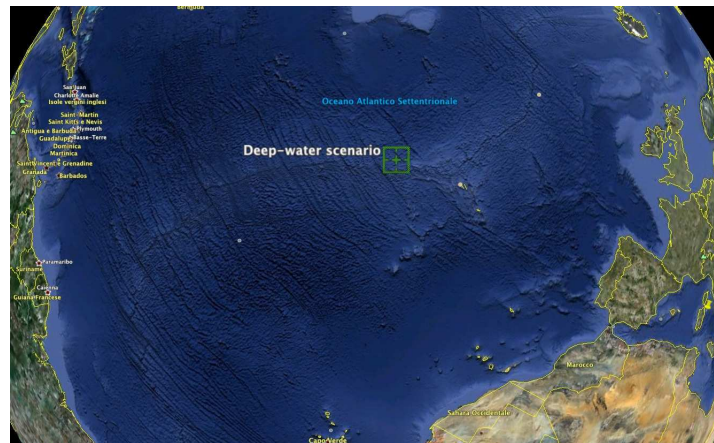


Figura 3.1: Scenario “deep-water” per le simulazioni.

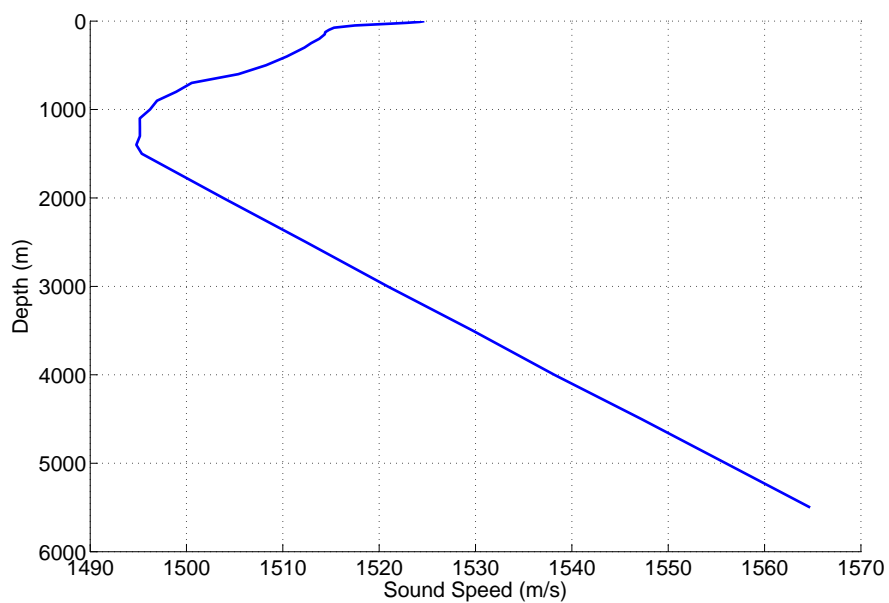


Figura 3.2: Sound Speed Profile estivo (Luglio) per la zona di Figura (3.1).

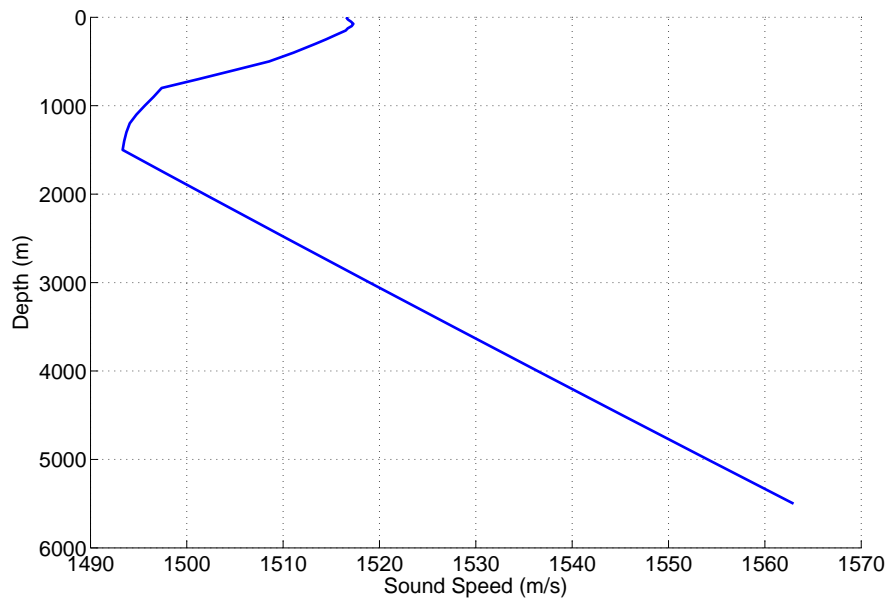


Figura 3.3: Sound Speed Profile invernale (Gennaio) per la zona di Figura (3.1).

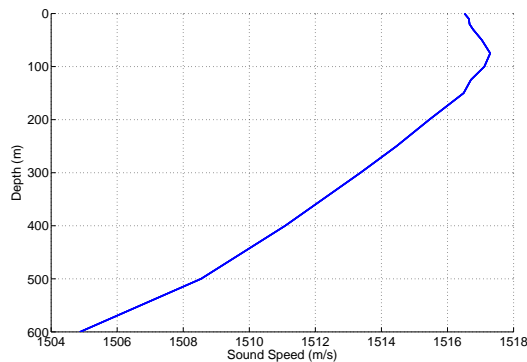


Figura 3.4: Particolare dell'SSP di Figura (3.3).

Per lo scenario *shallow-water*, invece, si è fatto riferimento al Mar Adriatico, nella zona dell'Istria, alle coordinate (**44.5 °N, 13.5 °E**) (Figura (3.5)). Le Figure (3.6) e (3.7) mostrano, invece, gli SSP correlati alla zona appena descritta. In questo caso la profondità del mare è decisamente minore rispetto a prima. Confrontando i due sound-speed-profile si nota come non ci sia un canale SOFAR così accentuato come nel caso precedente. Tuttavia, nel caso estivo, potrebbe essere conveniente trasmettere in profondità dove il suono ha una velocità minore. Il Sound Speed Profile invernale, invece, risulta essere quasi perfettamente rettilineo.

3. SIMULAZIONI E RISULTATI



Figura 3.5: Scenario “shallow-water” per le simulazioni.

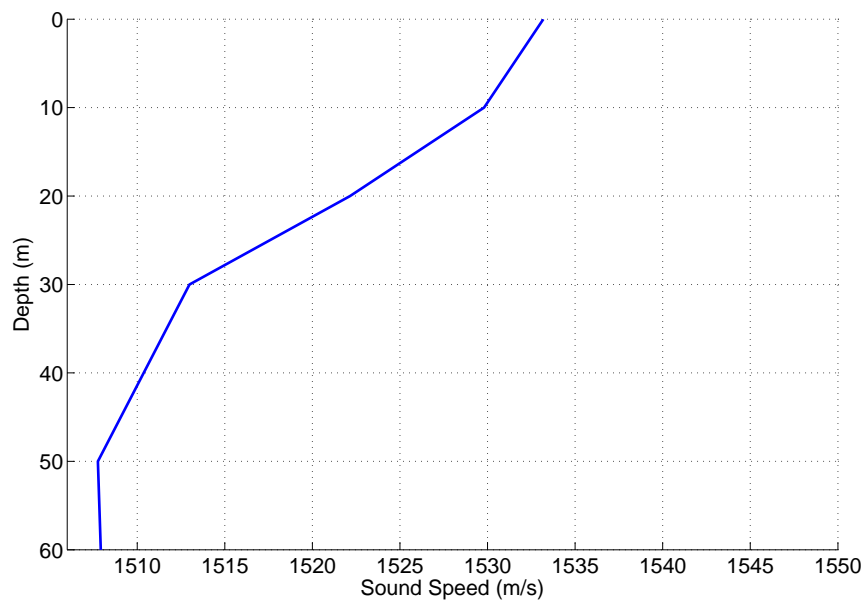


Figura 3.6: Sound Speed Profile estivo (Luglio) per la zona di Figura (3.5).

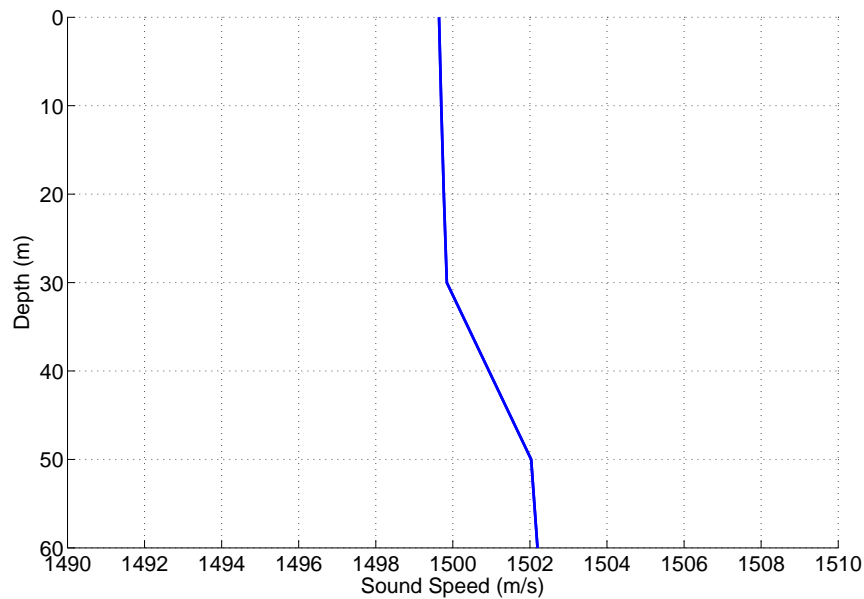


Figura 3.7: Sound Speed Profile invernale (Gennaio) per la zona di Figura (3.5).

3.1 Primo scenario : nodi in posizione casuale

3.1.1 Shallow-water

I nodi in questo caso sono disposti in maniera casuale uniforme all'interno di una zona di dimensioni di 2 km x 5 km. La profondità è invece scelta in maniera casuale tra 1 metro e 40 metri circa.

Le simulazioni sono state effettuate disponendo in maniera casuale questi nodi e selezionando 25 diverse realizzazioni di questo processo, e mediandone i risultati. L'effetto di protocolli superiori sui protocolli di accesso al mezzo in esame è stato simulato supponendo che ci sia un protocollo di routing che informa il nodo dei destinatari più vicini in maniera che solo i nodi collocati entro una certa distanza siano scelti come destinatari dei dati.

Ciò simula il fatto che questi nodi, in uno scenario più generale, potrebbero fare da relay per i messaggi mandati da un nodo a un destinatario fuori dal proprio raggio di copertura. In questo caso la rete è single-hop, quindi i nodi possono comunicare solo con i vicini.

Il raggio di copertura è calcolato in maniera che, in assenza di interferenza, il PDR (*Packet Delivery Ratio*) sia del 99 %, tenendo conto di avere la rete connessa, nel senso che esiste un percorso non necessariamente single-hop per raggiungere da

3. SIMULAZIONI E RISULTATI

un nodo qualsiasi altro nodo nella rete.

Qui di seguito sono riportati i parametri di livello fisico utilizzati per le simulazioni:

- **Periodo prescelto per la simulazione: Luglio (SSP di Figura 3.6)**
- **numero di nodi:** 20
- **Dimensione dei pacchetti dati:** 125 byte
- **Modulazione utilizzata:** BPSK
- **Frequenza della portante:** 25 kHz o 11.5 kHz
- **Bitrate:** 4800 bps
- **Banda:** 5 kHz
- **AcquisitionThreshold:** 5 dB
- **Potenza di trasmissione:** 130, 150 e 190 dB re μPa
- **Angolo di apertura del beam di trasmissione:**
180° (da -90° a +90°)

Con la configurazione sopra, il valore calcolato per la distanza di copertura è di **1000 m**.

Il parametro *AcquisitionThreshold* è un valore di SINR che permette di discriminare se il pacchetto è decodificabile correttamente o no. In particolare, se il segnale ricevuto ha un SINR minore di quella soglia, il pacchetto viene scartato direttamente dal livello fisico, altrimenti viene decodificato e mandato ai protocolli superiori. Questo valore è stato scelto abbastanza basso per rendere più robusti i nodi all'interferenza visto che la rete è molto densa. I pacchetti sono generati tramite un processo di Poisson ¹ in cui il parametro λ rappresenta il traffico della rete.

In Figura (3.8) e (3.9) si presentano i pattern di path loss costruiti con Bellhop alle frequenze portanti prescelte e indicate nella lista di parametri qui sopra. Nelle

¹Un processo di Poisson (abbr. PdP) è definito come $P[N(t + \tau) - N(t) = k] = \frac{\exp^{-\lambda\tau} (\lambda\tau)^k}{k!}$ dove $P[N(t + \tau) - N(t) = k]$ è la probabilità che nel tempo di τ il numero di eventi sia pari a k e λ viene definita intensità del processo.

figure si può osservare anche la batimetria della zona della simulazione ottenuta sempre da Bellhop tramite il file *.bty*.

Dai grafici si nota subito come le trasmissioni siano affette da molte riflessioni sul fondale e sulla superficie. Si nota, inoltre, la minore attenuazione presente alla frequenza $f = 11.5 \text{ kHz}$, in accordo con l'andamento del parametro $a(f)$ in Figura (1.1)

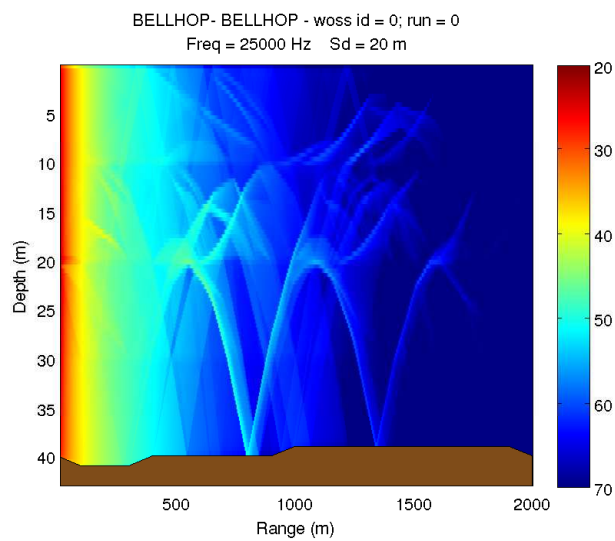


Figura 3.8: Pathloss di bellhop alla frequenza $f = 25 \text{ kHz}$.

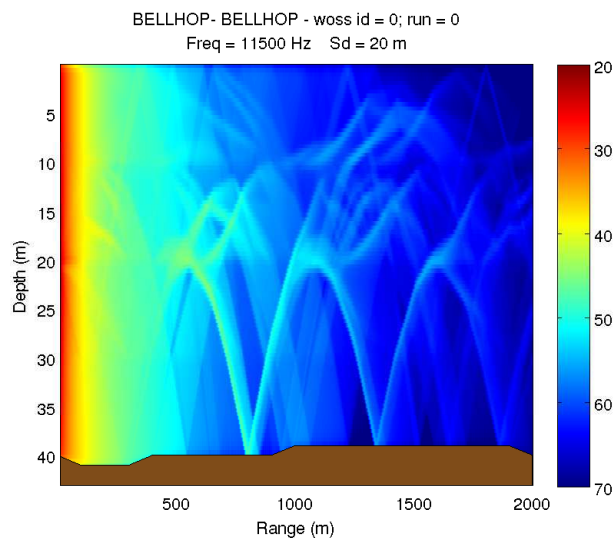


Figura 3.9: Pathloss di bellhop alla frequenza $f = 11.5 \text{ kHz}$.

3. SIMULAZIONI E RISULTATI

In Figura (3.10) viene presentato il throughput aggregato in relazione al traffico aggregato offerto nella rete per i vari protocolli per varie potenze e frequenze, mentre in Figura(3.11) è visibile il Packet Error Rate per nodo sempre in relazione al traffico aggregato nella rete.

Preliminarmente si può notare che alla frequenza portante di 25 kHz si ha un throughput più alto rispetto alla frequenza di 11.5 kHz; questo fatto è spiegabile ricordando i concetti espressi nel capitolo 1 di questa tesi. A 25 kHz, come si può vedere dalla Figura (1.2), abbiamo un contributo di rumore minore rispetto a 11.5 kHz, mentre alla frequenza più elevata abbiamo una attenuazione maggiore. Questo incremento di frequenza, aumenta da una parte l'attenuazione aiutando a diminuire l'interferenza presente nella rete (Figura (1.4)) e il contributo minore di rumore permette di avere un SNR al ricevitore sufficiente per avere delle prestazioni buone. Le differenze di throughput dovute a questo cambiamento di frequenza portante sono, nel caso peggiore del CSMA-ALOHA, dell'ordine dei 15 pck/min.

Prima di passare ad analizzare le prestazioni dei protocolli in esame si elencano i parametri utilizzati per questa simulazione:

- **Header:** 0 byte (il CSMA-ALOHA non necessita di overhead di informazioni)
- **Tempo di sensing del canale:** 0.5 s
- È stata considerata la versione senza ACK

Per quanto riguarda il DACAP, invece, abbiamo i seguenti parametri:

- **Tempo minimo per eseguire un Handshanke** t_{min} : 1.1842 s
- **Distanza ΔD :** 500 m
- **Massimo tempo di propagazione** max_prop_delay : 0.6579 s
- **HDR DACAP:** 10 byte
- **RTS:** 48 byte
- **CTS:** 48 byte
- **WRN:** 24 byte

Infine, per il protocollo TONE-LOHI si sono utilizzati i seguenti parametri

- **Massimo tempo di propagazione** max_prop_delay 0.6579 s
- **HDR TONE-LOHI:** 10 byte
- **Massimo numero di tentativi di trasmissione:** 1
- **recontend_time:** 15 ms (impostata a 3 volte la durata del tono per la contesa)
- **Durata del tono:** 5 ms
- **Versione di protocollo TONE-LOHI utilizzato:** aUT-Lohi

Il valore max_prop_delay è stato calcolato tenendo conto del fatto che i nodi trasmettono solo con i vicini a distanza di 1000 metri, come spiegato precedentemente, quindi,

$$max_prop_delay = \frac{1000}{1520} = 0.6579 \text{ s} \quad (3.1)$$

dove il valore 1520 m/s è stato preso come valore medio dall'ssp di Figura (3.6).

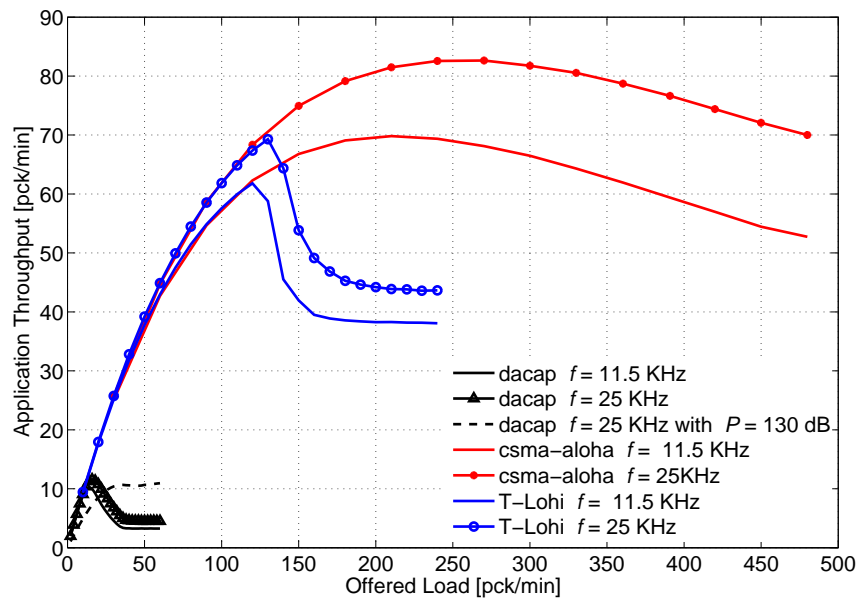


Figura 3.10: Throughput aggregato della rete.

Passando ad analizzare le prestazioni dei protocolli, i grafici mostrano che il CSMA-ALOHA ha delle prestazioni migliori sia in termini di throughput che di

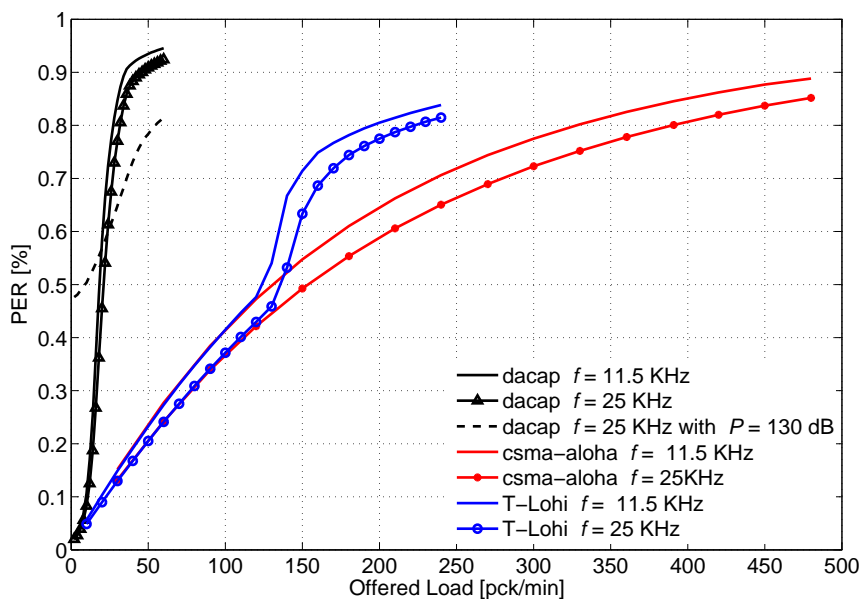


Figura 3.11: Packet Error Rate.

PER (Packet Error Rate).

Il TONE-LOHI ha prestazioni simili fino a un livello di traffico attorno ai 150 pck/min ma poi cala drasticamente, pur rimanendo costante su valori più bassi, aiutato dai time-out di backoff settati nel protocollo; il DACAP, invece, ha le prestazioni peggiori dei tre protocolli presi in considerazione.

In questo scenario, quindi, l'accesso casuale senza handshake è una politica vincente rispetto alle altre politiche adottate da DACAP e TONE-LOHI.

Il CSMA-ALOHA, infatti, ha dei bassi "tempi di settaggio", in cui il protocollo deve trasmettere dei messaggi di controllo o eseguire la contesa del canale, fatto salvo del tempo in cui effettua il sensing del canale per evitare alcune collisioni tra nodi vicini

Se ne conclude che trasmettere senza coordinare i nodi se non tramite il minimo coordinamento causato dal carrier sensing porta a delle prestazioni migliori rispetto a una politica di coordinazione tra i nodi.

DACAP, invece, controlla la trasmissione dei dati in maniera precisa tramite uno scambio di messaggi RTS/CTS. A causa dei tempi di propagazione elevati, i tempi morti in cui il protocollo si occupa di gestire la contesa del canale sono considerevoli, con conseguenze dirette sui valori del throughput.

D'altra parte, il valore di PER dimostra che comunque la elevata coordinazione

del DACAP non porta ad avere delle trasmissioni immuni dall'interferenza. Infatti, la Figura (3.11), dimostra che a livelli di traffico abbastanza elevati si ha un tasso di errore quasi del 100 %.

Questo dimostra che il DACAP soffre molto l'interferenza mutua tra i vari nodi e soprattutto mostra delle prestazioni pessime in reti molto dense di nodi. In questa rete, dove la zona è relativamente piccola, i nodi sono molto vicini tra di loro e sono quindi potenzialmente tutti interferenti tra di loro.

In particolare, i messaggi di controllo RTS e CTS, se mandati vicini a una trasmissione dati, hanno la capacità di interferire in maniera distruttiva con un messaggio dati. Si può concludere, quindi, che, quando la rete è sottoposta a un grande traffico dati, il DACAP non riesce a gestire l'interferenza, portando al fallimento quasi tutte le trasmissioni e mostrando un throughput molto basso.

Una soluzione possibile per risolvere questo problema è quella di alzare il valore di D , lasciando quindi che il messaggio RTS si propaghi più distante cercando di "render noto" a più nodi possibili il fatto della presenza di una imminente trasmissione. Innalzare questo parametro, tuttavia, porta il protocollo ad attendere più a lungo prima di eseguire una trasmissione dati.

Tutte queste considerazioni hanno portato ad eseguire una analisi più approfondita del DACAP in una situazione di basso traffico, cioè per carichi di rete minori rispetto a quelli considerati nella prima analisi.

Dalle Figure (3.12) e (3.13) si possono trarre varie conclusioni. Primariamente, si nota la mancanza di differenza in termini di throughput per le trasmissioni a 11.5 kHz e 25 kHz a una potenza trasmittiva di $P = 190dB \text{ re } \mu Pa$. Questo risultato va a confermare le conclusioni fatte precedentemente. Infatti, l'SNR del pacchetto ricevuto è sufficiente per decodificare correttamente il contenuto del messaggio.

Inoltre, si nota come il protocollo abbia delle prestazioni buone in termini di PER e throughput per valori di traffico nella rete relativamente bassi, che si attestano intorno a circa 10 pck/min aggregati nella rete. Questa frequenza di generazione di traffico lascia il tempo necessario al protocollo per eseguire correttamente la trasmissione senza che altre comunicazioni creino interferenza.

Si nota, inoltre, come innalzare la potenza non sia necessariamente sinonimo di migliorare le prestazioni. Infatti, settare la potenza a $P = 150dB \text{ re } \mu Pa$ porta a delle prestazioni migliori rispetto al caso a $P = 190dB \text{ re } \mu Pa$, seppure la

3. SIMULAZIONI E RISULTATI

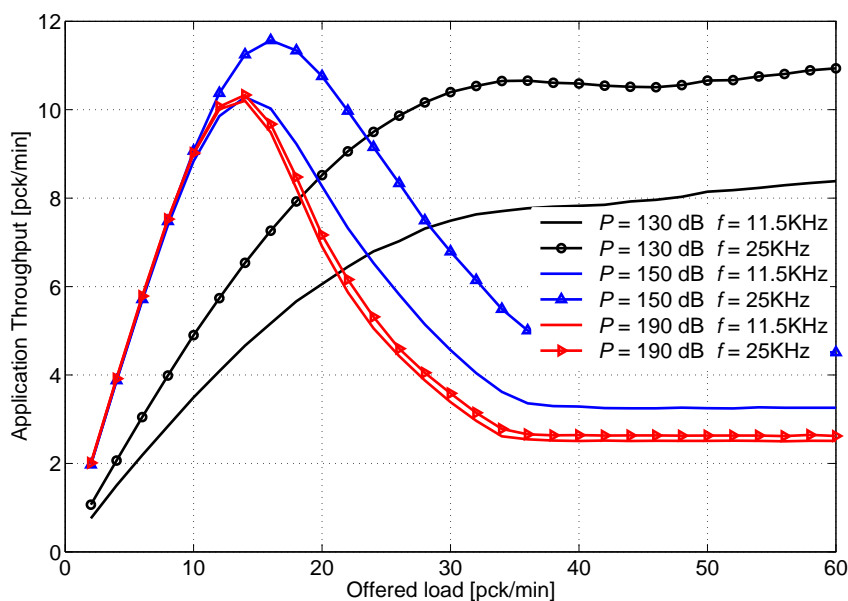


Figura 3.12: Throughput del protocollo DACAP con varie potenze trasmissive.

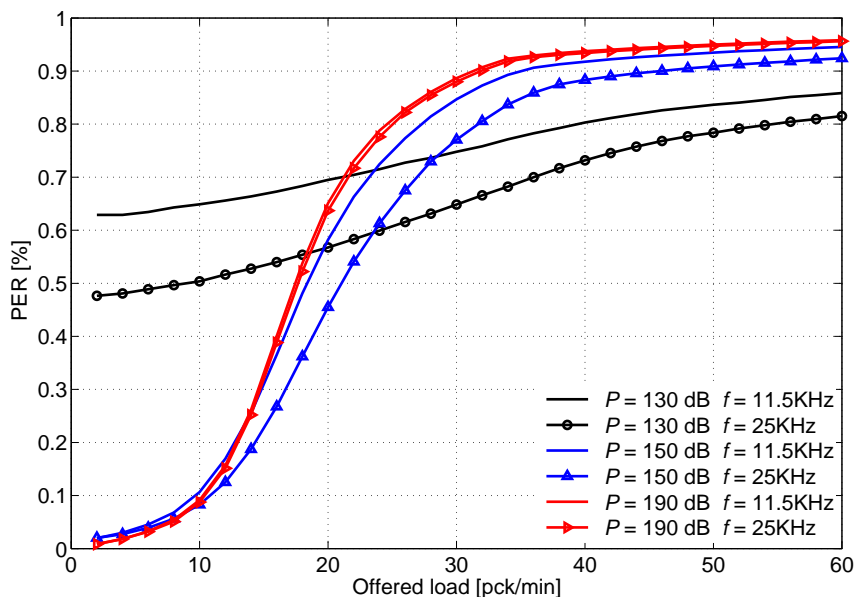


Figura 3.13: Packet Error Rate del protocollo DACAP con varie potenze trasmissive.

trasmissione a 11.5 kHz sia affetta da maggior rumore.

Addirittura, abbassare ulteriormente la potenza a $P = 130 \text{ dB re } \mu\text{Pa}$ porta a un throughput che cresce più a lungo rispetto agli altri casi, anche se il valore di picco non è comparabile con quello raggiunto trasmettendo a $P = 150 \text{ dB re } \mu\text{Pa}$.

Questo comportamento è il principale indicatore del fatto che il protocollo DACAP è pesantemente affetto dall'interferenza tra le trasmissioni dei vari nodi. Abbassare la potenza trasmissiva significa raggiungere con un SNR minore i nodi più lontani, ma significa anche ridurre l'interferenza con i nodi circostanti, perchè la trasmissione copre solo una zona ridotta con una potenza tale da poter essere dannosa per una eventuale altra comunicazione. Questo effetto è comunque mitigato se il traffico nella rete è molto basso.

Infatti, se il tasso di generazione del traffico è molto basso, il protocollo ha tutto il tempo per instaurare la connessione e trasmettere i dati prima che un altro pacchetto venga generato e quindi si inneschi la procedura di handshake per la trasmissione; in questo caso, quindi, il problema principale nella rete è l'SNR, che non è sempre sufficientemente elevato per raggiungere le distanze decise, supponendo di aver abbassato la potenza al di sotto del valore per cui è stata calcolata la distanza di copertura con un Packet Error Rate del 99 %. Appena però il traffico cresce, l'interferenza tra le trasmissioni cresce di conseguenza, e il problema principale nella rete diventa il SIR. In questo caso, quindi, una potenza trasmissiva relativamente bassa quale $P = 130dB \text{ re } \mu Pa$ può avvantaggiare la comunicazione.

Ovviamente, dalla Figura (3.12) si evince che con la potenza trasmissiva più bassa non si raggiunge mai un valore di throughput di picco raggiunto nel caso della potenza di $P = 150dB \text{ re } \mu Pa$. Dipende, però dall'applicazione specifica; se la rete è progettata per lavorare a traffici molto bassi, conviene alzare la potenza per avere un throughput massimo più alto. Se però il traffico nella rete è destinato a crescere, è bene avere una potenza trasmissiva che permette di avere un throughput accettabile anche per livelli di traffico più elevati.

Per quanto riguarda il valore di ΔD , si è ritenuta necessaria una adeguata simulazione per trovare un valore ottimo. Le Figure (3.14) e (3.15) sono state ottenute fissando il traffico della rete a un valore alto (50 pck/min) alla potenza fissa di $P = 190dB \text{ re } \mu Pa$ in maniera da riuscire a capire se la variazione di ΔD immunizza effettivamente le trasmissioni dei pacchetti dall'interferenza.

Dai grafici si può subito notare come aumentare il valore ΔD non porti a miglioramenti e, anzi, porti a un leggero peggioramento delle prestazioni.

Per quanto riguarda il throughput, il peggioramento delle prestazioni può essere spiegato ricordando che innalzando il valore di ΔD si va ad aumentare il tempo

3. SIMULAZIONI E RISULTATI

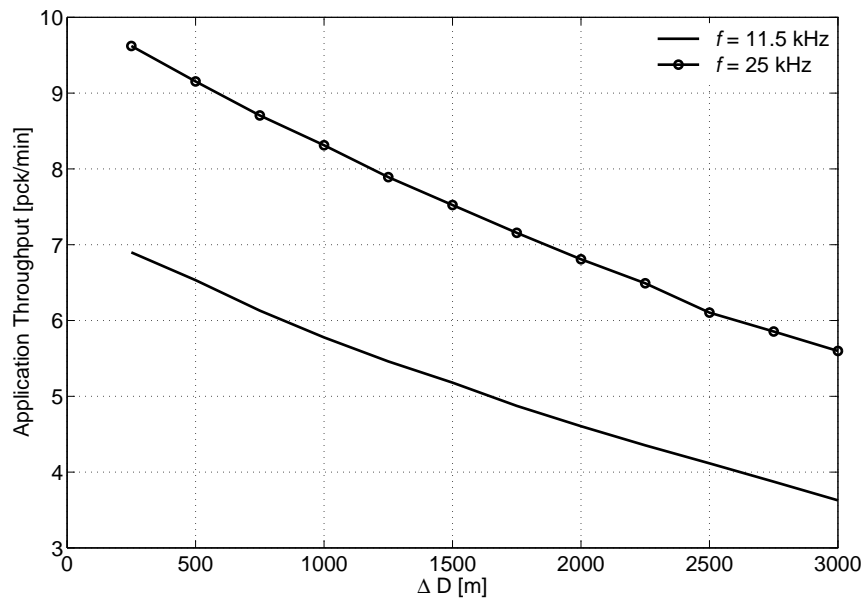


Figura 3.14: Throughput per il protocollo DACAP al variare del valore di ΔD .

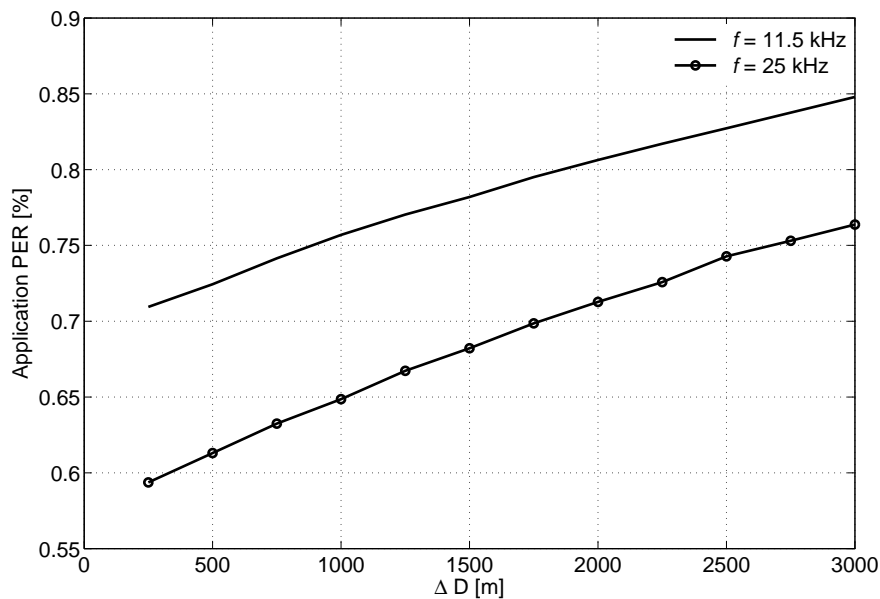


Figura 3.15: Packet Error Rate dovuto all'interferenza per il protocollo DACAP al variare del valore di ΔD .

di warning, dedicando meno tempo alle trasmissioni utili di pacchetti dati.

Per il Packet Error Rate, tuttavia, il risultato presentato in Figura (3.15) può essere contro-intuitivo. Il fatto di innalzare il valore di ΔD , diminuisce il tempo dedicato dal protocollo alla trasmissione dati, ma il tempo di vulnerabilità dov-

to al ritardo di propagazione del canale rimane. Perciò, nelle simulazioni è stato scelto un valore piccolo (500 metri).

Un altro parametro che può contribuire alla variazione delle prestazioni nel DACAP è la distanza a cui un nodo può essere considerato “vicino” da un altro nodo (quindi può essere destinatario delle sue trasmissioni).

Nelle Tabelle (3.1) e (3.2) possiamo vedere come effettivamente una distanza di copertura minore porti a delle prestazioni migliori per il protocollo DACAP.

Questo miglioramento non è dovuto solamente al fatto che la tratta di propagazione è minore e quindi l'SNR ricevuto è migliore, ma anche al fatto che i tempi dedicati all'handshaking sono sensibilmente minori e l'RTT diminuisce e, quindi, il tempo di vulnerabilità del protocollo diminuisce concordemente.

3. SIMULAZIONI E RISULTATI

Frequenza portante: 11.5 kHz.		
Distanza di copertura [m]	Throughput [pck/min]	PDR
1000	5.31	0.22
1500	4.19	0.17
2000	2.39	0.09

Tabella 3.1: Throughput e PDR utilizzando il protocollo DACAP per differenti distanze di copertura. Il traffico è settato a 50 pck/min e la potenza è $P = 150dB$ re μPa

Frequenza portante: 25 kHz.		
Distanza di copertura [m]	Throughput [pck/min]	PDR
1000	5.98	0.26
1500	4.78	0.2
2000	3.27	0.13

Tabella 3.2: Throughput e PDR utilizzando il protocollo DACAP per differenti distanze di copertura. Il traffico è settato a 50 pck/min e la potenza è $P = 150dB$ re μPa .

Il protocollo TONE-LOHI, invece, ha delle prestazioni medie tra il DACAP e il CSMA-ALOHA. Per traffici non molto alti, ma comunque maggiori del DACAP, il TONE-LOHI ha prestazioni praticamente uguali al CSMA-ALOHA, segno che la strategia della trasmissione del tono per la contesa e il backoff dei nodi proporzionale al numero di nodi che hanno partecipato alla contesa è vincente e riesce ad avere prestazioni molto simili al CSMA-ALOHA.

Tuttavia, per traffici più alti, quando cioè la rete comincia a essere congestionata, questa tecnica non è più efficiente per vincere l'interferenza presente nella rete. Questo fatto è visibile in Figura (3.11), quando, per valori di traffico aggregato nella rete che variano dai 100 ai 150 pck/min, il PER ha un improvviso peggioramento.

Analoghe argomentazioni valgono per il throughput (Figura 3.10) in cui la curva di throughput comincia a scendere per gli stessi valori di traffico per poi saturare a un valore più basso mentre il throughput del CSMA-ALOHA continua a salire. Quando il traffico è elevato nella rete, la maggior parte del tempo viene speso nella contesa e di conseguenza gli slot di CR (Contention Round) sono necessariamente più lunghi perchè molti più nodi vi partecipano e tutte le trasmissioni presenti nella rete restituiscono un livello di interferenza che il protocollo non riesce a gestire.

La strategia del TONE-LOHI è senza dubbio più vincente di quella del DACAP, perchè il tono trasmesso per iniziare una contesa va a interferire meno con le trasmissioni dati già presenti sul canale, rispetto a un pacchetto RTS.

Tuttavia, anche i toni che vengono trasmessi molto spesso in condizioni di alto traffico, possono interferire con la trasmissione dei dati utili.

In questo caso, comunque, non si è ritenuta necessaria un'analisi più approfondita del protocollo perchè le prestazioni sono comunque buone almeno per valori medi di traffico nella rete, e le ragioni di queste prestazioni sono facilmente comprensibili.

Passiamo ora ad analizzare le prestazioni del protocollo USB-AIMD, paragonandolo al protocollo CSMA-ALOHA puro, con lo scopo di vedere se la politica di ARQ adottata porta a un miglioramento delle prestazioni. Il protocollo USB-AIMD è stato configurato come segue:

- **Tempo di guardia:** 0.1 s
- **Modo ARQ:** Selective Repeat train

3. SIMULAZIONI E RISULTATI

- **Transmission Mode:** “PD”

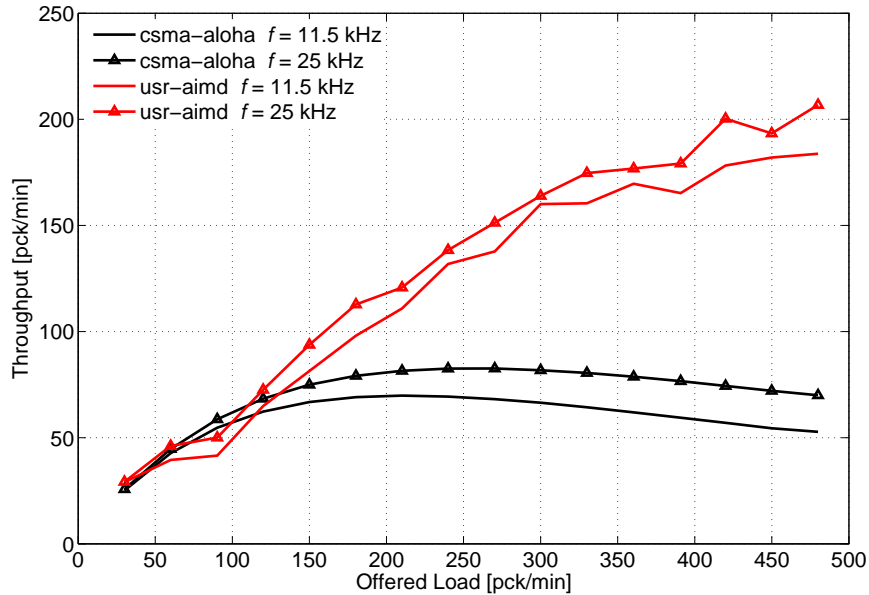


Figura 3.16: Confronto throughput tra CSMA-ALOHA e USR-AIMD .

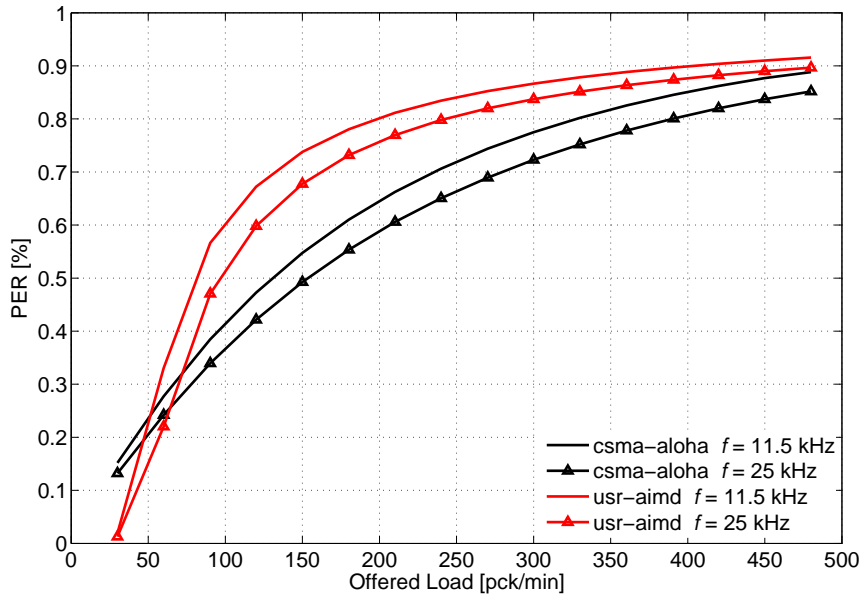


Figura 3.17: Confronto Packet Error Rate tra CSMA-ALOHA e USR-AIMD .

In Figura (3.16) mostra il throughput del protocollo USR-AIMD in relazione a quello del CSMA-ALOHA, mentre la Figura (3.17).

Possiamo notare come il protocollo USB esibisca un throughput migliore del CSMA-ALOHA “puro” ma, al contario, dopo un certo valore di traffico, abbia un PER leggermente peggiore.

Il throughput maggiore è la conseguenza di un utilizzo migliore nel canale. Infatti, l’USB-AIMD riesce a trasmettere più pacchetti in un tempo di propagazione, riuscendo, quindi, tramite la finestra AIMD a sfruttare meglio la capacità del canale in quel momento. Inoltre, l’interallacciamento intelligente di ricezione degli ACK e trasmissione dei pacchetti dati, riesce a ottimizzare i tempi dedicati alla trasmissione, con effetti benefici sul throughput.

Quando la rete è scarica, la politica di ritrasmissione permette di ottenere PER nullo, ma, all’aumentare del traffico, l’interferenza diventa più grande e le collisioni più probabili. Viene quindi richiesto un numero sempre maggiore di ritrasmissioni che vanno a saturare sempre di più il canale.

Ecco quindi spiegato il motivo di un PER peggiore al caso CSMA-ALOHA. Resta tuttavia il fatto che, pur avendo un tasso di errore peggiore, il numero di pacchetti nell’unità di tempo consegnati correttamente è maggiore rispetto al CSMA-ALOHA per i motivi appena spiegati.

3.1.2 Deep-water

Anche in questo scenario abbiamo 20 nodi disposti in maniera casuale uniforme in un’area più grande rispetto a quella presa in considerazione per lo scenario shallow.

L’area considerata in questo caso è di 5 km x 20 km; per quanto riguarda, invece, la profondità, i nodi sono disposti in maniera aleatoria a una profondità variabile tra 1000 e i 2000 metri.

Osservando l’SSP, queste profondità corrispondono al *SOFAR channel*. Esiste, quindi, la possibilità di verificare il contributo del SOFAR channel nella propagazione del segnale acustico a grandi distanze.

In Figura (3.18) e (3.19) abbiamo i pattern di propagazione creati da bellhop per le due frequenze di lavoro prescelte, in entrambe le figure è evidente l’effetto del SOFAR channel alla profondità media di 1500 metri. In questo caso non è presente la batimetria della zona, perchè i nodi sono flottanti, la profondità del mare è molto maggiore (circa 4000-5000 metri) e la batimetria stessa viene considerata ininfluyente per la comunicazione. Anche in questo ambiente le simulazioni

3. SIMULAZIONI E RISULTATI

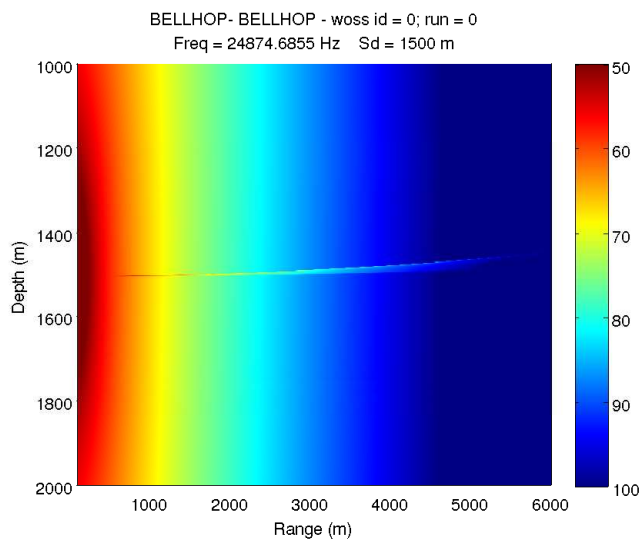


Figura 3.18: Pattern di propagazione restituito da Bellhop nella zona prescelta alla frequenza di 25 kHz .

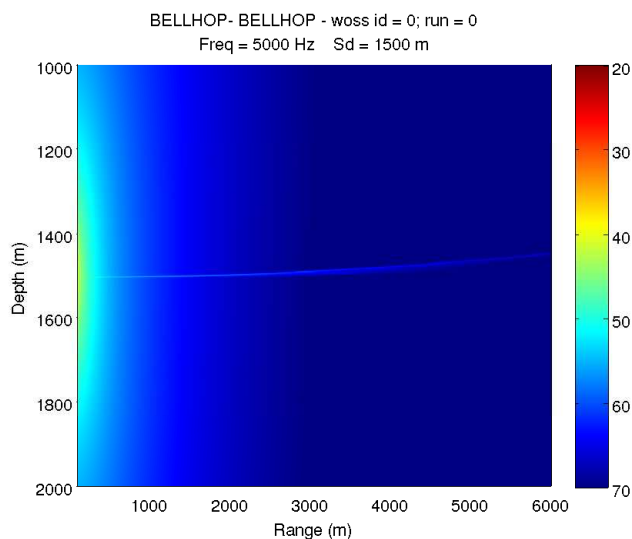


Figura 3.19: Pattern di propagazione restituito da Bellhop nella zona prescelta alla frequenza di 5 kHz. Si può notare che l'attenuazione massima in questo caso è minore (Figura (3.18))

vengono effettuate su 25 diverse topologie e i risultati mediati tra di loro. Anche in questo ambiente si è decisa una distanza massima a cui i nodi trasmettono i loro dati. In questo, caso, essendo la rete più estesa rispetto al caso precedente, abbiamo un raggio di copertura di **5000 metri**. Questo valore, significativamente maggiore rispetto al precedente, è stato scelto per garantire che

la rete sia connessa.

Essendo la zona molto più estesa rispetto al caso precedente, la rete risulta molto più sparsa e, anche aumentando il raggio di copertura, i nodi “vicini” per ciascun nodo non aumentano sensibilmente per ogni nodo.

Andiamo a elencare anche in questo caso i parametri utilizzati per le simulazioni

- **Periodo prescelto per la simulazione:** Luglio (SSP di Figura 3.2)
- **numero di nodi:** 20
- **Dimensione dei pacchetti dati:** 125 byte
- **Modulazione utilizzata:** BPSK
- **Frequenza della portante:** 25 kHz o 5 kHz
- **Bitrate:** 4800 bps o 800 bps
- **Banda:** 5 kHz
- **AcquisitionThreshold:** 5 dB
- **Potenza di trasmissione:** 150 e 190 dB re μPa
- **Angolo di apertura del beam di trasmissione:**
180° (da -90° a +90°)

dove il significato dei vari parametri è spiegato nel paragrafo precedente. Inizialmente le simulazioni sono state eseguite con una frequenza di 5 kHz a un bitrate di 800 bps e successivamente innalzando sia il bitrate che la frequenza portante (4800 bps e 25 kHz), per verificare se il tempo minore di trasmissione dei pacchetti dati porta a un vantaggio maggiore rispetto all'aumento dell'attenuazione dovuto alla maggiore frequenza portante.

Anche in questo ambito il traffico dei nodi è demandato a un modulo CBR che genera dati secondo un processo di Poisson.

I parametri del CSMA-ALOHA sono invariati rispetto al caso precedente, mentre i parametri del DACAP e del TONE-LOHI sono variati per le condizioni ambientali variate (per esempio il maggior tempo massimo di propagazione aumentato).

- **Tempo minimo per eseguire un Handshake t_{min} :** 6.4 s

3. SIMULAZIONI E RISULTATI

- **Distanza ΔD):** 500 m
- **Massimo tempo di propagazione max_prop_delay :** 3.33 s
- **HDR DACAP:** 10 byte
- **RTS:** 48 byte
- **CTS:** 48 byte
- **WRN:** 24 byte

Invece, per il protocollo TONE-LOHI abbiamo i seguenti parametri

- **Massimo tempo di propagazione max_prop_delay** 6.4 s
- **HDR TONE-LOHI:** 10 byte
- **Massimo numero di tentativi di trasmissione:** 1
- **recontend_time:** 15 ms (impostata a 3 volte la durata del tono per la contesa)
- **Durata del tono:** 5 ms
- **Versione di protocollo TONE-LOHI utilizzato:** aUT-Lohi

Il massimo tempo di propagazione è stato calcolato come rapporto tra il raggio massimo di copertura (5000 metri) e la velocità del suono scelta per default a 1500 m/s .

il valore t_{min} prescelto per il DACAP, invece, è stato scelto in maniera che $t_{min} < 2T$, partendo dal presupposto che, come nel caso shallow, i nodi non siano al bordo della zona di propagazione ma siano mediamente più vicini. In Figura (3.22) e (3.20) ci sono i throughput dei protocolli testati alle due frequenze di $f = 5 kHz$ e $f = 25 kHz$.

Anche in questo environment il protocollo CSMA-ALOHA raggiunge le prestazioni migliori, seguito da TONE-LOHI e DACAP.

Come nel caso shallow, le prestazioni del CSMA-ALOHA sono spiegabili ricordando che l'area dove i nodi sono disposti è molto vasta. I nodi sono sufficientemente distanti tra di loro e le trasmissioni riescono ad avvenire in maniera disaccoppiata. Per questo motivo, ridurre la latenza del canale e la mancanza di pattern di

segnalazione che occupano inutilmente il canale risulta la strategia vincente. Il TONE-LOHI, invece, presenta un throughput minore a causa dei numeri tempi di attesa imposti dal protocollo. Il DACAP, da parte sua, ha un ulteriore svantaggio dettato dall'interferenza tra nodi vicini. Anche un piccolo messaggio di controllo come l'RTS o il CTS può essere nocivo per una trasmissione dati vicina. Abbiamo quindi che il tempo per effettuare l'handshaking viene sprecato e la trasmissione viene persa perchè corrotta dall'interferenza.

Tra l'altro, nell'ambiente deep i nodi sono mediamente più distanti tra di loro e quindi il RTT medio aumenta; aumenta di conseguenza anche il tempo di vulnerabilità del protocollo. Le collisioni tra i pacchetti di controllo e i dati diventano quindi più probabili e distruttive, portando alle prestazioni limitate riportate nelle Figure.

Tuttavia, in Figura (3.21) possiamo notare come il DACAP abbia un PER migliore rispetto al TONE-LOHI; la coordinazione richiesta dal DACAP, pur soffrendo dei punti deboli di cui si è appena parlato, permette di avere meno collisioni rispetto al TONE-LOHI, però i tempi di attesa sono molto maggiori quindi il throughput risulta minore.

Osservando la Figura (3.20) possiamo notare, inoltre, come innalzare la potenza da $190 \text{ dB re } \mu Pa$ a $200 \text{ dB re } \mu Pa$ non porti a dei sostanziali miglioramenti nelle prestazioni. Questo significa che la potenza più bassa è sufficiente per coprire la tratta e vincere sia l'attenuazione che il rumore presente sul canale e che il problema maggiore in questo ambito è proprio l'interferenza mutua tra le varie comunicazioni. In questo senso, quindi, aumentare la potenza significa aumentare l'interferenza senza ottenere un significativo aumento delle prestazioni.

A questo punto la frequenza portante è stata innalzata a $f = 25 \text{ kHz}$ con lo scopo di vedere come i protocolli reagiscono all'aumento dell'attenuazione.

Osservando le differenze tra la Figura (3.22) e la Figura (3.20) e tra la Figura (3.23) e la Figura (3.21) si può notare come le prestazioni del CSMA-ALOHA migliorino mentre quelle del DACAP e del TONE-LOHI peggiorino leggermente. Il motivo delle migliori prestazioni del CSMA-ALOHA è da ascrivere sempre ai grandi delay dovuti ai tempi di propagazione. Inoltre, questi tempi di attesa sono diversi tra i vari nodi e questo contribuisce a disaccoppiare le trasmissioni tra i vari nodi che in questo modo non collidono. Anche con una attenuazione aumentata a causa della frequenza più alta, i pacchetti arrivano a destinazione con un buon

3. SIMULAZIONI E RISULTATI

SNR, con il contributo anche della propagazione facilitata del canale SOFAR; infatti, anche alzando la potenza a $200 \text{ dB re } \mu\text{Pa}$ le prestazioni non migliorano DACAP e TONE-LOHI, da un lato, traggono beneficio dall'incremento dell'attenuazione del canale alla suddetta frequenza, a causa della minore interferenza presente globalmente sulla rete. Questo fatto è dimostrato in Figura (3.23) in cui DACAP e TONE-LOHI hanno dei PER leggermente migliori rispetto al caso precedente. Inoltre, il DACAP ha un throughput più stabile per traffici maggiori rispetto al caso a frequenza minore.

Tuttavia, questa attenuazione aggiuntiva che da un lato protegge dall'interferenza, rende, dall'altro, più difficoltoso il "set-up" del canale: infatti, i messaggi di controllo per il DACAP (RTS e CTS) e i toni per quanto riguarda il TONE-LOHI giungono a destinazione con un SNR minore. I tempi di backoff e i tempi di attesa rendono così il throughput basso.

Come nel caso precedente, si è voluto investigare l'andamento delle prestazioni

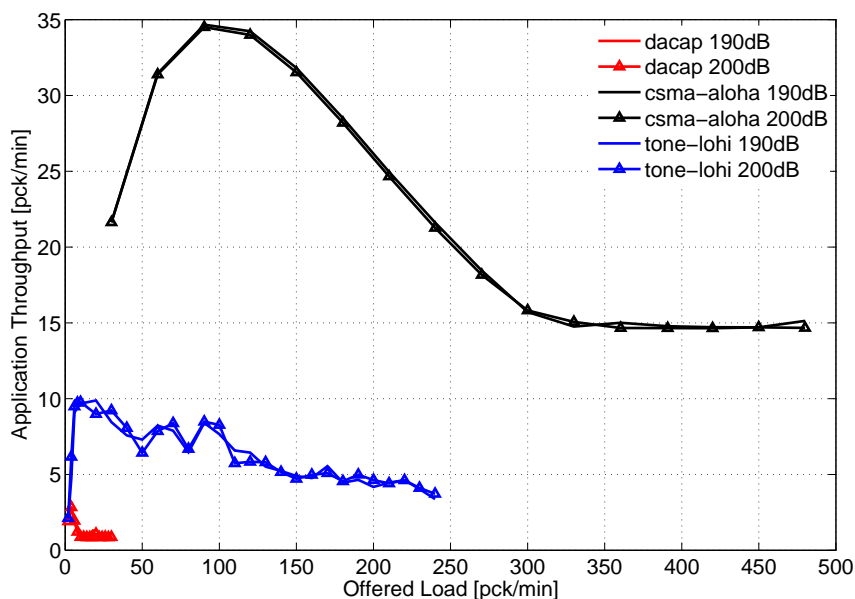


Figura 3.20: Throughput nella rete in funzione del traffico offerto per una frequenza di $f = 5 \text{ kHz}$.

nel protocollo DACAP in funzione della variazione dei parametri, in particolare del parametro ΔD . Come nel caso shallow, le prestazioni non necessariamente migliorano aumentando questo parametro e, anzi, peggiorano leggermente (Figura (3.24)) a causa dei grandi tempi di attesa dovuti al tempo di warning T_w che

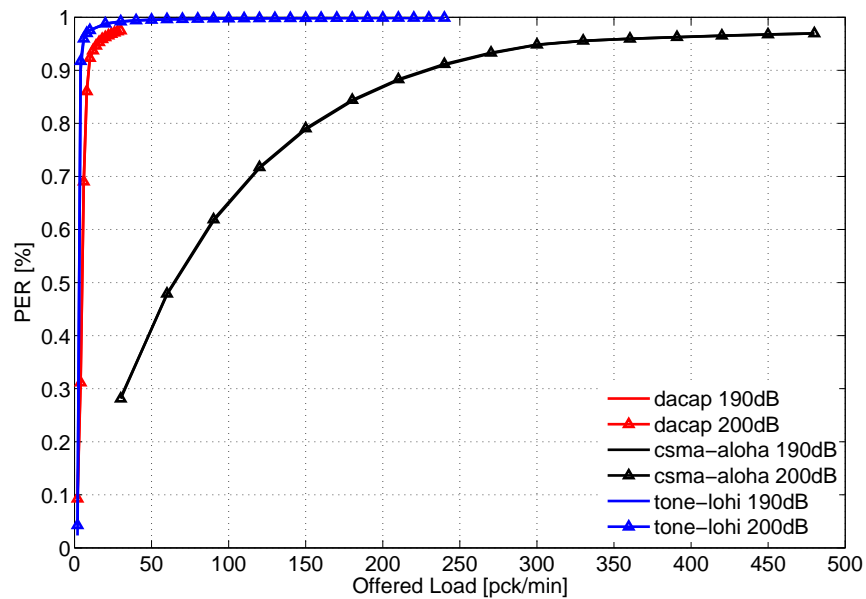


Figura 3.21: Packet Error Rate in funzione del traffico offerto per una frequenza di $f = 5 \text{ kHz}$.

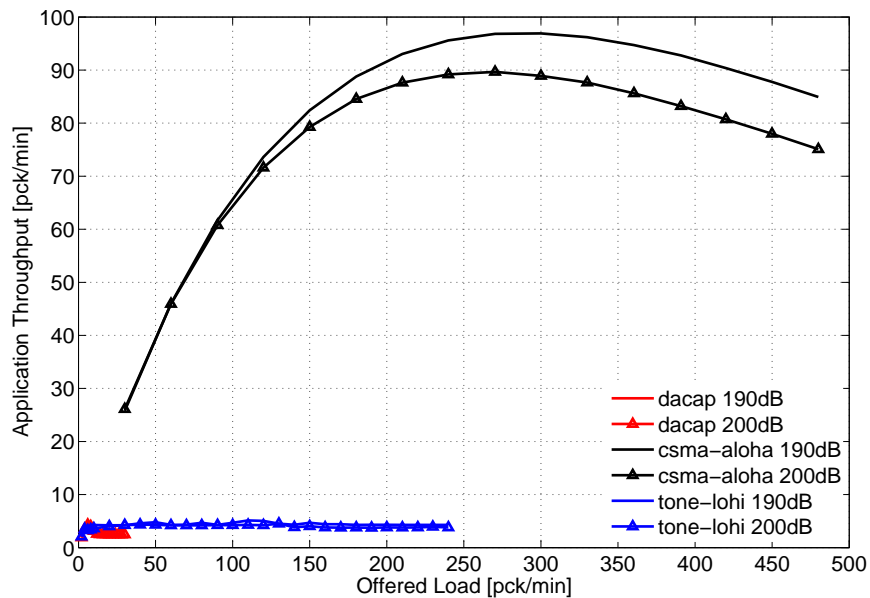


Figura 3.22: Throughput nella rete in funzione del traffico offerto per una frequenza di $f = 25 \text{ kHz}$.

aumenta concordemente a ΔD .

Possiamo comunque notare come nel caso deep water il throughput presenti una

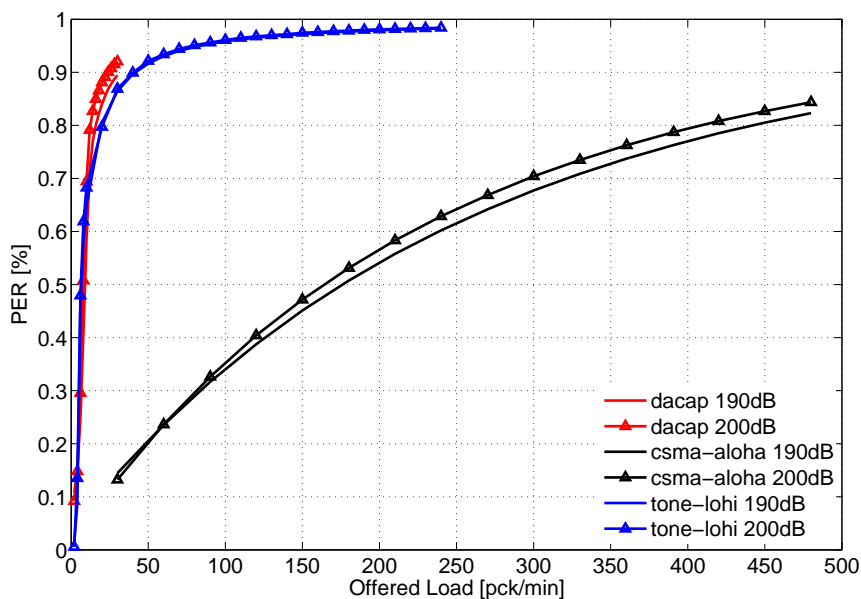


Figura 3.23: Packet Error Rate nella rete in funzione del traffico offerto per una frequenza di $f = 25 \text{ kHz}$.

diminuzione poco pronunciata rispetto al caso shallow.

Questo effetto è intuitivo perchè l'aumento dei tempi di attesa dovuti alla variazione, anche sostanziale, del parametro in esame è contenuto se confrontato coi tempi di propagazione in una rete molto sparsa come quella considerata.

In una rete più densa (caso shallow water), la variazione dei tempi di attesa è invece comparabile coi tempi di propagazione e quindi il throughput ne risente direttamente.

L'ultima osservazione molto importante da fare per quanto riguarda la Figura (3.24), è che le prestazioni alla potenza di $190 \text{ dB re } \mu\text{Pa}$ sono migliori, conferma ulteriore che tenere la potenza più bassa permette di tenere l'interferenza sotto controllo. Alla fine di questa sezione presentiamo il confronto tra il protocollo CSMA-ALOHA e il protocollo USR-AIMD. Anche nell'ambiente deep si possono ripetere i ragionamenti e le conclusioni tratti per l'ambiente shallow. Abbiamo che l'USR-AIMD ha delle performance sostanzialmente maggiori rispetto al CSMA-ALOHA puro in termini di throughput. Queste prestazioni migliori sono da ricercare, quindi, nel fatto che l'USR-AIMD riesce a gestire meglio gli RTT trasmettendo più pacchetti e riesce ad interallacciare la ricezioni degli ACK con la trasmissione del pacchetto.

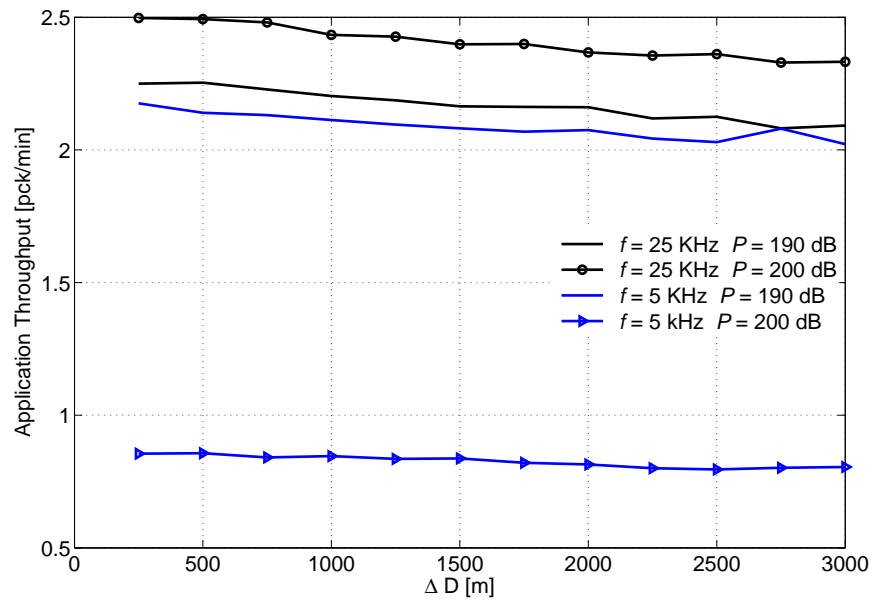


Figura 3.24: Throughput nella rete in funzione del parametro ΔD per un traffico di 50 pck/min.

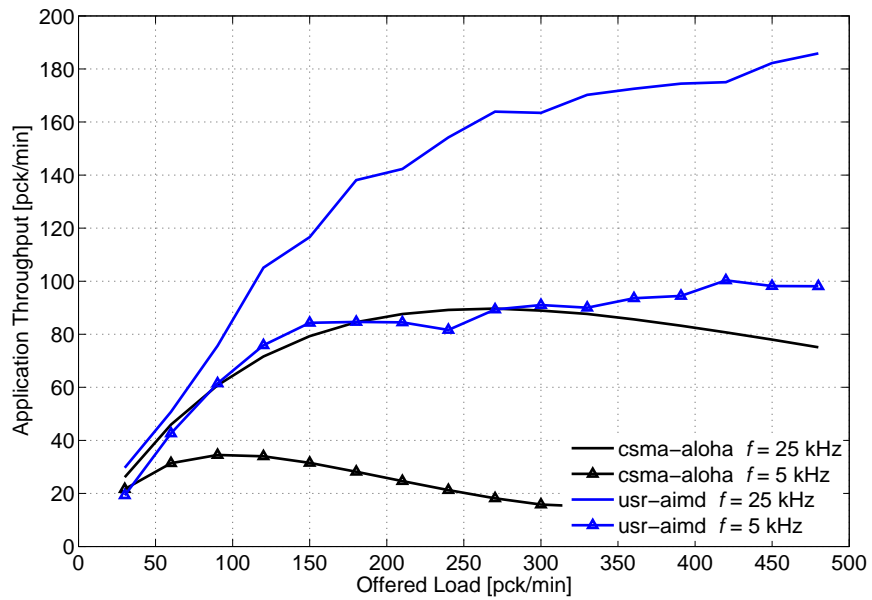


Figura 3.25: Throughput nella rete in funzione del traffico offerto per i protocolli USR-AIMD e CSMA-ALOHA.

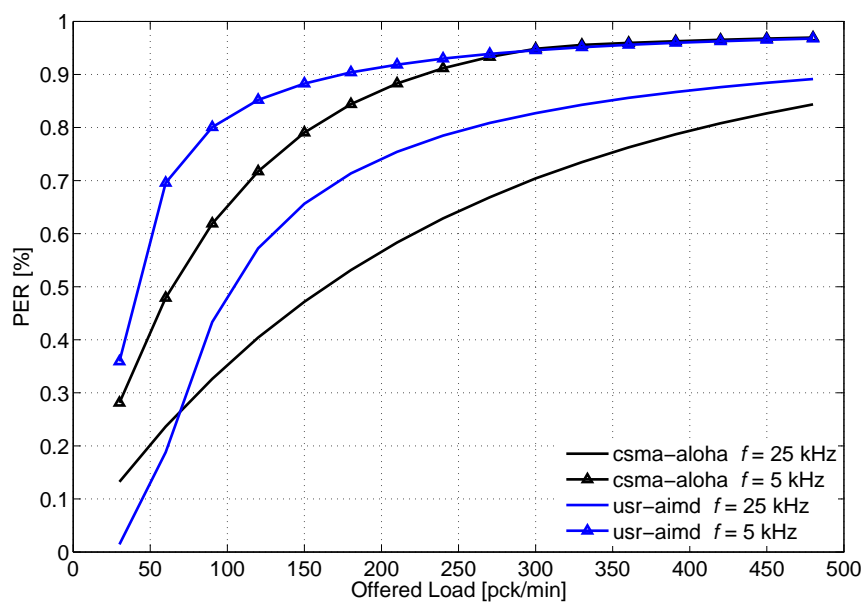


Figura 3.26: PER nella rete in funzione del traffico offerto per i protocolli USR-AIMD e CSMA-ALOHA.

3.2 Secondo scenario: nodi in posizione fissa che comunicano con un sink superficiale

In questa seconda parte prenderemo in considerazione un disposizione diversa dei nodi per verificare se questi cambiamenti vanno ad influenzare le prestazioni del protocollo e in quale maniera.

In questo caso abbiamo 8 nodi disposti, in due file da 4 nodi ciascuna in cui ogni nodo dista dai suoi vicini di 1 km. I nodi trasmetteranno tutti a un unico sink disposto al centro di questa rete e ad una profondità di 5 metri (si suppone che il sink sia una boa sulla superficie con il ricevitore sommerso).

La profondità dei nodi è variabile nei due casi considerati.

3.2.1 Shallow-water

La zona prescelta per la simulazione è la stessa del caso precedente e i nodi, in questo caso, sono disposti sul fondale.² I parametri del livello fisico prescelti per le simulazioni eseguite sono i seguenti:

- **Periodo prescelto per la simulazione:** Luglio (SSP di Figura 3.6)
- **numero di nodi:** 8 (disposti a griglia)
- **Dimensione dei pacchetti dati:** 125 byte
- **Modulazione utilizzata:** BPSK
- **Frequenza della portante:** 25 kHz
- **Bitrate:** 4800 bps
- **Banda:** 5 kHz
- **AcquisitionThreshold:** 5 dB
- **Potenza di trasmissione:** 150 dB re μPa

²Per la precisione, i nodi sono disposti a una profondità di 30 m e non sono effettivamente appoggiati al fondo. Dalla batimetria si può notare come il fondale sia “frastagliato”, si è scelta una profondità media in maniera da avere una rete planare. I nodi, comunque, si possono considerare ancorati al fondo.

3. SIMULAZIONI E RISULTATI

- **Angolo di apertura del beam di trasmissione:**

180° (da -90° a +90°)

La potenza è stata calcolata in maniera che tutti i nodi possano trasmettere al sink separatamente con un PDR del 99 %. Quindi, la potenza è sufficiente per fare in maniera che ogni nodo riesca a trasmettere dati con un SNR (considerando path loss calcolato da Bellhop e rumore) tale per cui i pacchetti possano essere sempre decodificati correttamente in assenza di interferenza.

I parametri dei protocolli sono sostanzialmente uguali al caso di disposizione di nodi random, a parte i valori che tengono conto delle distanze tra il nodo e il sink (in questo caso), per il DACAP abbiamo quindi:

- **Tempo minimo per eseguire un Hand-Shaking t_{min} :** 1.9 s
- **Distanza ΔD):** 500 m
- **Massimo tempo di propagazione max_prop_delay :** 1.05 s
- **HDR DACAP:** 10 byte
- **RTS:** 48 byte
- **CTS:** 48 byte
- **WRN:** 24 byte

e per il TONE-LOHI:

- **Massimo tempo di propagazione max_prop_delay** 1.05 s
- **HDR TONE-LOHI:** 10 byte
- **Massimo numero di tentativi di trasmissione:** 1
- **recontend_time:** 15 ms (impostata a 3 volte la durata del tono per la contesa)
- **Durata del tono:** 5 ms
- **Versione di protocollo TONE-LOHI utilizzato:** aUT-Lohi

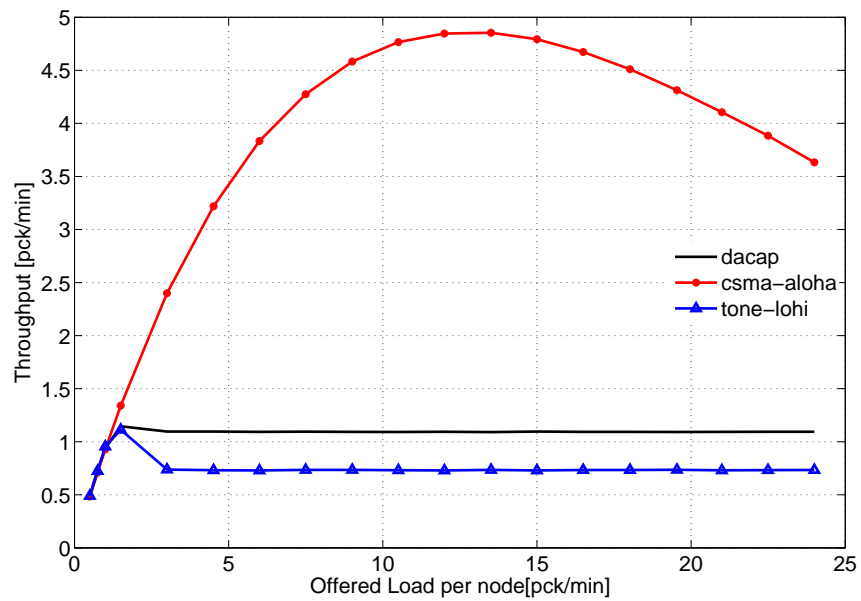


Figura 3.27: Confronto di CSMA-ALOHA, TONE-LOHI e DACAP. Throughput per nodo in funzione del traffico offerto.

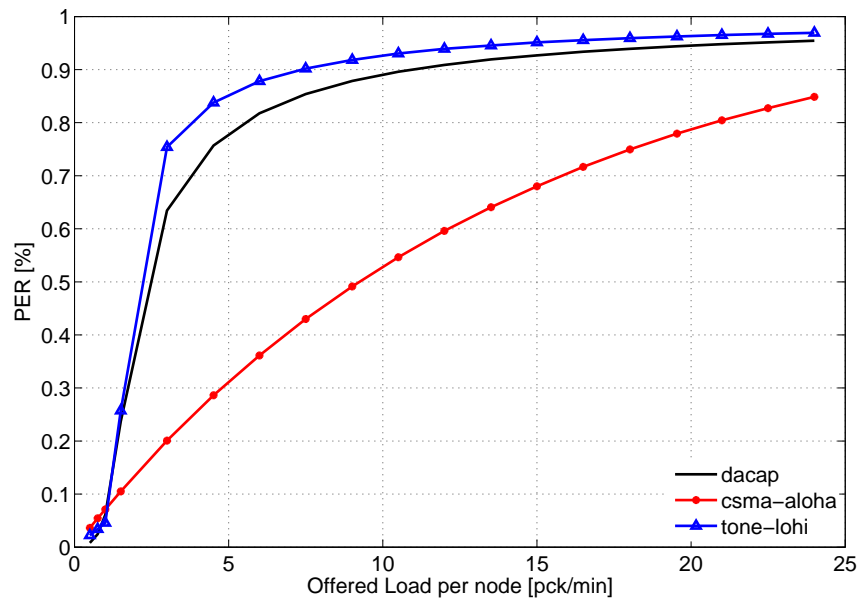


Figura 3.28: Confronto di CSMA-ALOHA, TONE-LOHI e DACAP. Packet Error Rate in funzione del traffico offerto.

3. SIMULAZIONI E RISULTATI

Il massimo ritardo di propagazione è stato calcolato calcolando la distanza fisica tra il nodo più distante (i nodi agli angoli della rete) e il sink.

Per il protocollo DACAP, il valore t_{min} è stato invece calcolato come il doppio del ritardo di propagazione tra il sink e il nodo più vicino a esso.

Nelle Figure (3.27) e (3.28) si riportano rispettivamente il throughput e il Packet Error Rate dei tre protocolli presi esame. Ancora una volta, come il caso precedente, il CSMA-ALOHA risulta il protocollo che ha le performances migliori sia in termini di throughput che di tasso di errore.

Rispetto al caso precedente, osserviamo che a traffici molto bassi (molto minori di 5 pck/s per nodo) i tre protocolli hanno performances molto simili e, in termini di PER, i protocolli che adottano una sorta di coordinazione tra i nodi (DACAP e TONE-LOHI) riescono ad avere delle prestazioni migliori.

Quest'ultima osservazione rappresenta la differenza più evidente rispetto al caso precedente, dove il CSMA-ALOHA vinceva anche in condizioni di rete "scarica".

In questo caso, l'interferenza della rete non è distribuita su tutta la dimensione della rete, ma è centralizzata al sink, destinatario di tutte le comunicazioni dei nodi, che può controllare l'interferenza cercando una coordinazione tra i nodi (scambio di RTS/CTS). Inoltre, le dimensioni della rete stessa sono tali per cui, supponendo che un pacchetto si propaghi in ogni direzione, trasmettere, ad esempio, un RTS (prendendo a campione il protocollo DACAP) da parte di un nodo al sink significa "rendere noto" a più di metà dei nodi componenti la griglia suddetta il fatto che a breve si il canale sarà occupato da una trasmissione. La rete può essere coperta in maniera quasi completa settando opportunamente il parametro ΔD (anche se successivamente dimostreremo che questo parametro non avrà una grande influenza nelle prestazioni).

Queste considerazioni sono valide, però, solo se passa un tempo sufficientemente alto tra l'inizio della trasmissione di un RTS e la generazione di un altro pacchetto da parte di un altro nodo. Questo significa che, se il tasso di generazione (quindi il traffico), è maggiore, la supposizione appena fatta non è più valida perchè ci saranno più RTS o CTS simultanei mandati, che possono collidere tra di loro, oppure collidere con pacchetti dati ed essere dannosi per la trasmissione, come notato anche nella rete precedente. Inoltre, il sink si vedrà recapitare con crescente frequenza RTS da parte di nodi che vogliono trasmettere e, seguendo la politica dettata dal DACAP, dovrà avvisare gli altri nodi con cui aveva iniziato

lo scambio RTS/CTS di annullare la trasmissione tramite un pacchetto WRN. Inoltre, i nodi stessi, captando molti CTS e RTS che si propagheranno nella rete, saranno costretti ad annullare la trasmissione.

Innalzare il traffico nella rete, quindi, per i motivi appena evidenziati, porta il DACAP ad avere prestazioni peggiori rispetto all'accesso casuale al canale fornito dal CSMA-ALOHA.

In merito al confronto da TONE-LOHI e DACAP, si nota come il DACAP abbia delle prestazioni leggermente migliori rispetto al TONE-LOHI.

Nel caso del TONE-LOHI, si osserva come il traffico faccia aumentare il tempo dedicato ai *Contention Round* (CR), a causa dei numerosi nodi coinvolti che partecipano al CR. Questi tempi di "setting-up" vanno a ripercuotersi direttamente sul throughput. Tuttavia, le differenze tra TONE-LOHI e DACAP sono modeste.

Come nel caso precedente è stata effettuata un'analisi del parametro ΔD del protocollo DACAP, per approfondire, con questa disposizione di nodi, il suo ruolo e verificare se le ipotesi poste precedentemente trovano una conferma anche in questo caso. Il traffico è stato fissato a 4 pck/min per nodo. Le Figure (3.29) e (3.30) evidenziano come il parametro ΔD non porti dei benefici se innalzato. A parte delle oscillazioni locali di modesta entità, i valori sia di throughput che di PER sono sostanzialmente costanti.

Per valori alti di ΔD , si aspetta che il pacchetto di controllo si propaghi sostanzialmente in tutta la rete. In questo lasso di tempo, però, molti altri nodi stanno accedendo al canale mandando il loro RTS e CTS, a causa del tasso di generazione relativamente elevato. Come è noto dalle considerazioni nella rete precedente, tuttavia, trattare valori di ΔD molto elevati porta anche ad avere dei tempi di set-up molto più elevati, abbassando di conseguenza il throughput.

Concludendo, si può affermare che le prestazioni buone del protocollo CSMA-ALOHA sono confermate anche in uno scenario in cui abbiamo dei nodi che trasmettono con un unico sink superficiale. Inoltre, possiamo osservare che i protocolli DACAP e TONE-LOHI, che richiedono una certa coordinazione tra i nodi, migliorano, seppur di poco, le loro prestazioni. Questo, sostanzialmente, significa che riescono a trattare in maniera migliore l'interferenza subita da un solo ricevitore rispetto all'interferenza distribuita uniformemente nella rete in cui i nodi sono sparsi in maniera casuale.

3. SIMULAZIONI E RISULTATI

Inoltre, anche in questo caso è evidente che la modifica e l'eventuale aumento del parametro ΔD non porta effettivamente a un miglioramento delle prestazioni, confermando le ipotesi fatte nel caso precedente.

Convieni, quindi, tenere questo parametro ad un valore molto basso in maniera da diminuire il più possibile i tempi del protocollo e cercando, quindi, un throughput più alto.

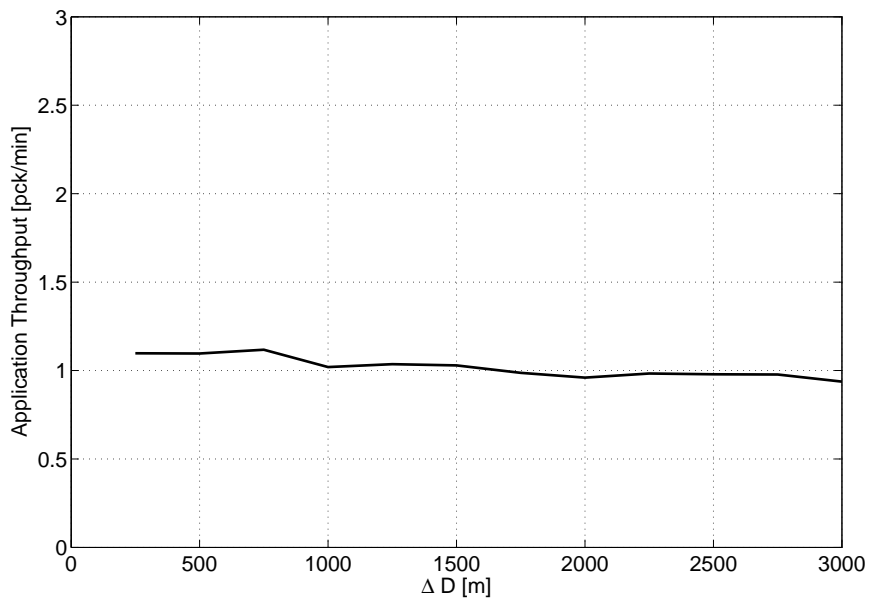


Figura 3.29: Throughput per nodo in funzione del parametro ΔD (Il traffico è fissato a 4 pkt/min per nodo)

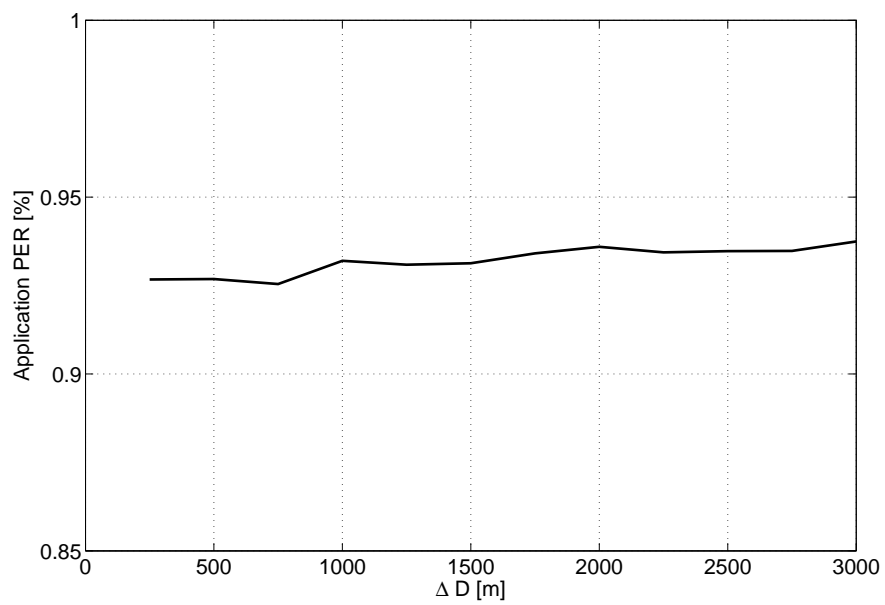


Figura 3.30: Packet Error Rate per nodo in funzione del parametro ΔD (Il traffico è fissato a 4 pkt/min per nodo)

3. SIMULAZIONI E RISULTATI

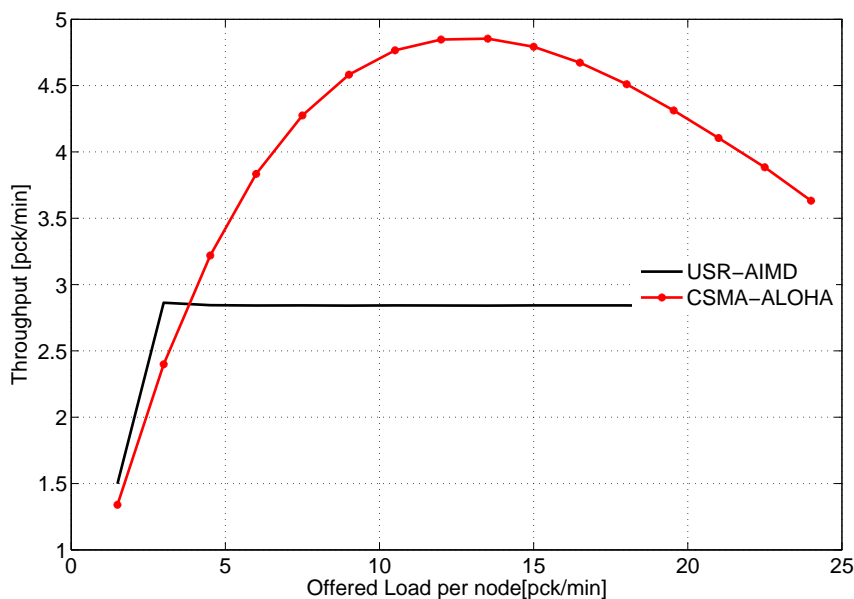


Figura 3.31: Throughput per nodo dei protocolli CSMA-ALOHA e USR-AIMD in funzione del traffico offerto per nodo.

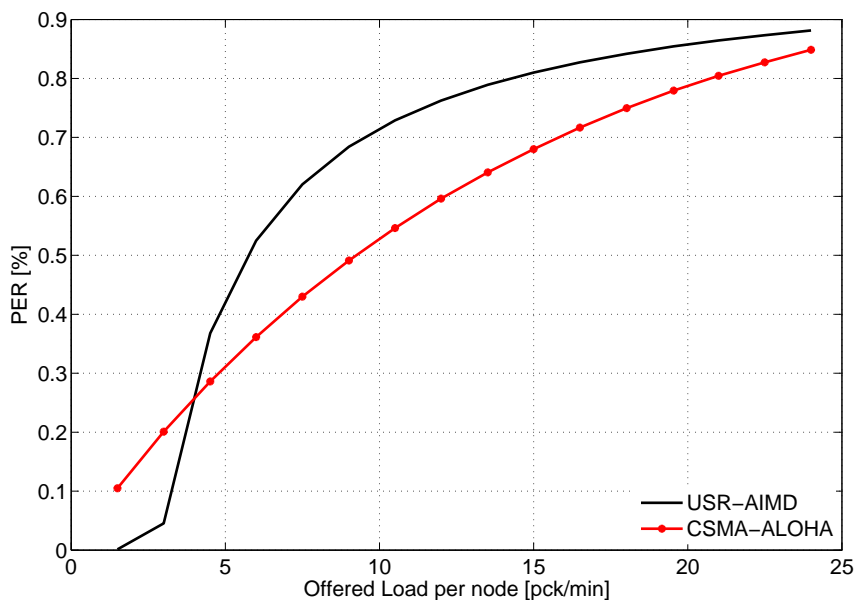


Figura 3.32: Packet Error Rate per nodo dei protocolli CSMA-ALOHA e USR-AIMD in funzione del traffico offerto per nodo.

Le Figure (3.31) e (3.32) presentano le prestazioni del protocollo USR-AIMD, confrontato con il protocollo CSMA-ALOHA.

In questo caso, le differenze rispetto al caso dei nodi disposti in maniera aleatoria sono evidenti. Il PER presenta un comportamento simile al caso precedente. Per valori di traffico molto bassi, la tecnica di ritrasmissione è molto efficace e riesce a mantenere il tasso di errore della rete pressochè nullo. Quando però il traffico cresce, la politica di ARQ non riesce a smaltire le sempre crescenti richieste di ritrasmissione dovute a collisioni più frequenti, con il conseguente peggioramento del Packet Error Rate rispetto a una politica CSMA-ALOHA semplice.

La differenza più evidente rispetto alla disposizione casuale dei nodi sta nel throughput. Se nel caso precedente, pur avendo un PER peggiore, la tecnica utilizzata da ARQ per utilizzare al meglio il canale permetteva di avere un throughput sempre crescente, ora questa tecnica non riesce ad avere le stesse prestazioni.

La differenza principale tra il caso precedente e il caso in esame è che in questo caso abbiamo una interferenza concentrata al sink. Le comunicazioni non sono disaccoppiate nello spazio e avvengono molte collisioni al ricevitore. La finestra AIMD, quindi, a causa delle continue collisioni, non riesce ad espandersi, portando il throughput a saturare a un valore costante. Il sistema di acknowledgement e l'adattabilità del protocollo alle diverse situazioni (cambio di politica di ARQ da SR a S&W e variazione del numero di pacchetti mandati in un RTT dinamica) permettono, tuttavia, di far saturare il throughput a un valore medio.

Possiamo quindi concludere che adottare una politica di ARQ con una rete in cui tutti i nodi devono comunicare concorrentemente a un unico destinatario non è la scelta vincente e sicuramente adottare una tecnica di accesso al canale casuale come il CSMA-ALOHA porta a delle prestazioni migliori sia in termini di PER che di throughput. Sarebbe invece una possibile correzione se si potesse assegnare un "turno" di trasmettere a ciascun nodo, in modo da evitare l'interferenza al sink. Questo richiede modifiche sostanziali al protocollo MAC, che saranno discusse nel prossimo capitolo.

3.2.2 Deep-water

La disposizione dei nodi in questo scenario è analoga alla precedente, con l'unica differenza che in questo caso i nodi si suppongono fluttuanti a una profondità di 100 metri.

I parametri prescelti per il livello fisico sono praticamente uguali al caso shallow-

3. SIMULAZIONI E RISULTATI

water, ad eccezione della potenza trasmissiva, che è stata alzata per fare in maniera, come prima, che tutti i nodi possano trasmettere pacchetti con PDR al 99 % al sink. Riportiamo, comunque, i parametri, per completezza:

- **Periodo prescelto per la simulazione:** Luglio (SSP di Figura 3.6)
- **numero di nodi:** 8 (disposti a griglia)
- **Dimensione dei pacchetti dati:** 125 byte
- **Modulazione utilizzata:** BPSK
- **Frequenza della portante:** 25 kHz
- **Bitrate:** 4800 bps
- **Banda:** 5 kHz
- **AcquisitionThreshold:** 5 dB
- **Potenza di trasmissione:** 190 dB re μPa
- **Angolo di apertura del beam di trasmissione:**
180° (da -90° a +90°)

Per quanto riguarda i parametri del DACAP e del TONE-LOHI, i parametri sono stati adattati alla nuova distanza (aumentata) tra nodi e sink, nel caso del DACAP abbiamo:

- **Tempo minimo per eseguire un Hand-Shaking t_{min} :** 1.9 s
- **Distanza ΔD):** 500 m
- **Massimo tempo di propagazione max_prop_delay :** 1.05 s
- **HDR DACAP:** 10 byte
- **RTS:** 48 byte
- **CTS:** 48 byte
- **WRN:** 24 byte

mentre per il TONE-LOHI:

- **Massimo tempo di propagazione** *max_prop_delay* 1.05 s
- **HDR TONE-LOHI:** 10 byte
- **Massimo numero di tentativi di trasmissione:** 1
- **recontend_time:** 15 ms (impostata a 3 volte la durata del tono per la contesa)
- **Durata del tono:** 5 ms
- **Versione di protocollo TONE-LOHI utilizzato:** aUT-Lohi

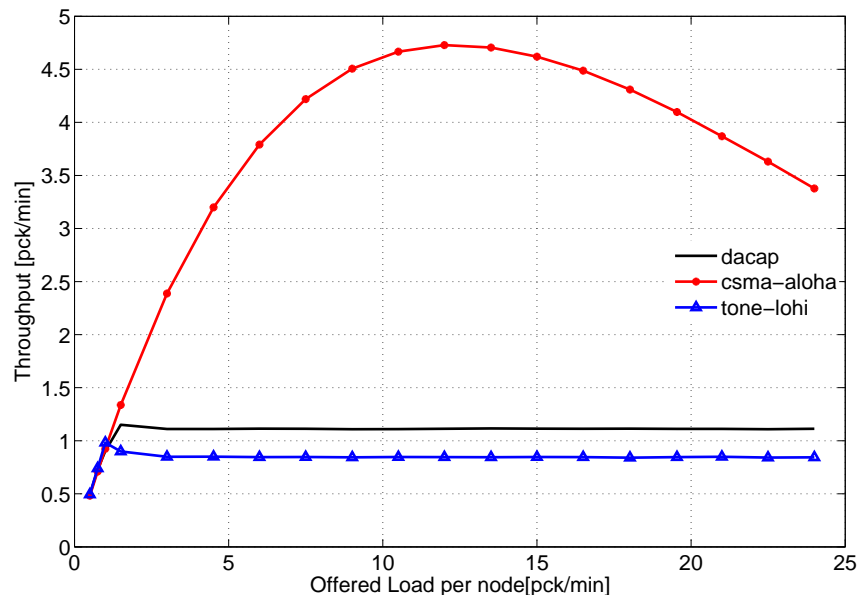


Figura 3.33: Confronto di CSMA-ALOHA, TONE-LOHI e DACAP. Throughput per nodo in funzione del traffico offerto.

Anche in questo caso le Figure (3.33) e (3.34) riportano il throughput e il PER, rispettivamente, nel caso di ambiente deep-water.

Le considerazioni che si possono fare sono, in sostanza, uguali a quelle fatte nella sezione precedente per il caso shallow-water. Il comportamento dei protocolli, infatti è praticamente uguale e quindi continuano a valere le stesse ipotesi del caso precedente. Anche per il parametro ΔD del protocollo DACAP (Figure (3.35) e (3.36)) si possono fare le medesime considerazioni fatte nel caso precedente.

3. SIMULAZIONI E RISULTATI

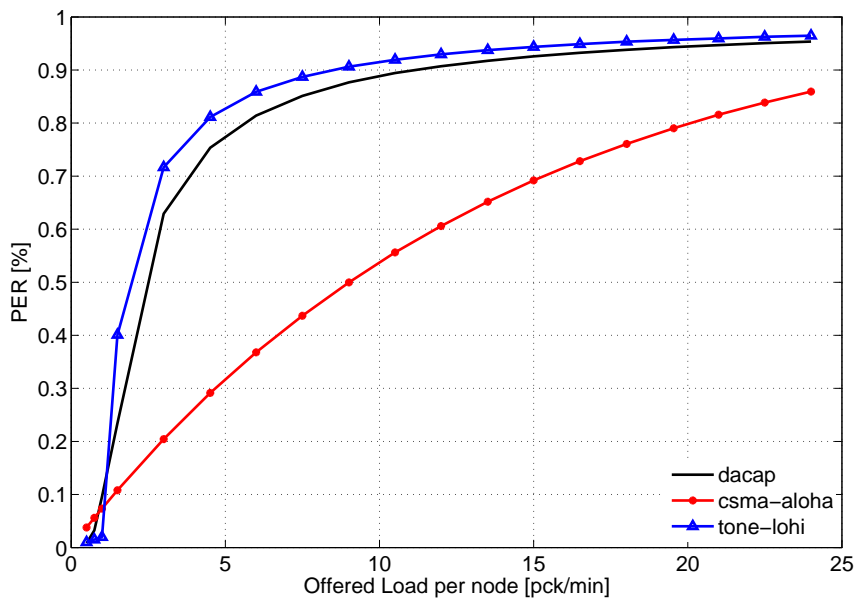


Figura 3.34: Confronto di CSMA-ALOHA, TONE-LOHI e DACAP. Packet Error Rate in funzione del traffico offerto.

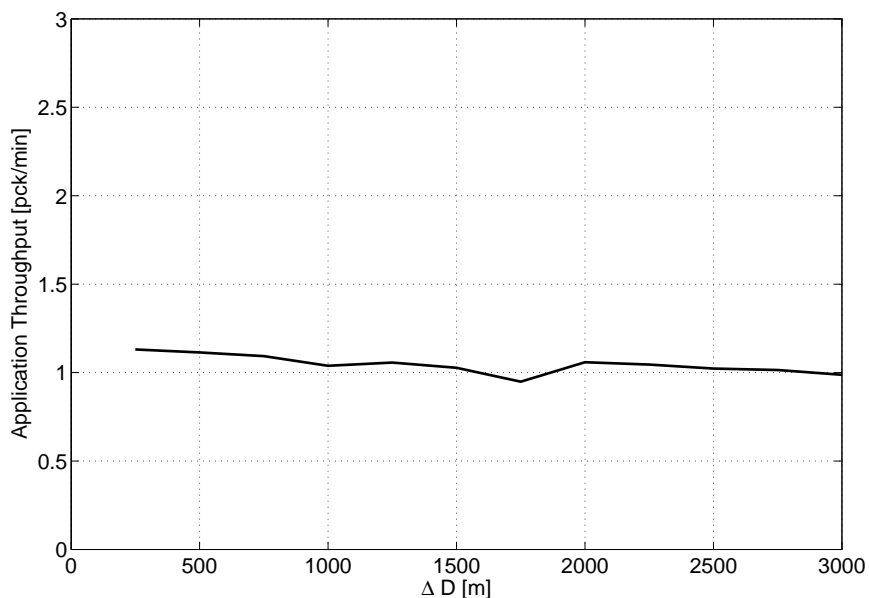


Figura 3.35: Throughput per nodo in funzione del parametro ΔD . (Il traffico è fissato a 4 pkt/min per nodo)

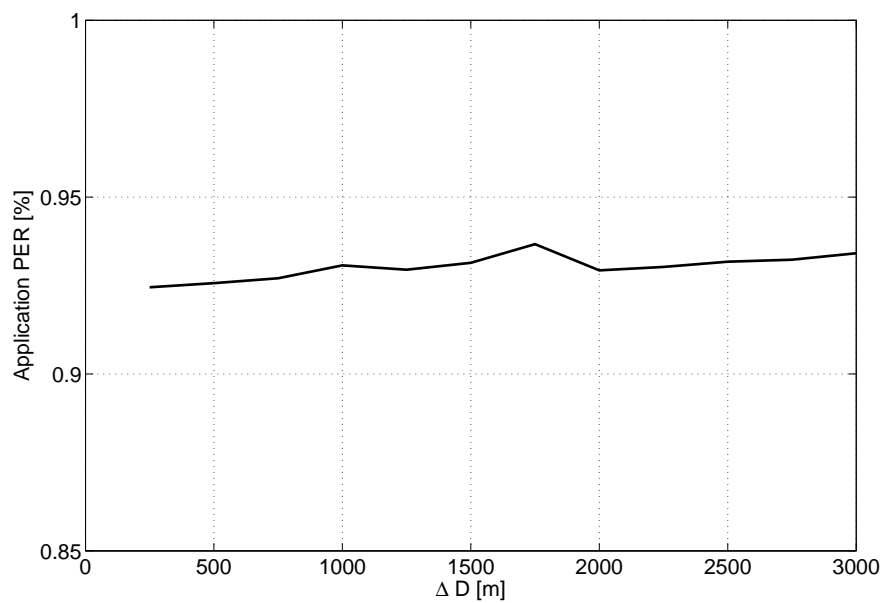


Figura 3.36: Packet Error Rate per nodo in funzione del parametro ΔD . (Il traffico è fissato a 4 pkt/min per nodo)

3. SIMULAZIONI E RISULTATI

In questo scenario, invece, si vuole mettere in evidenza un aspetto finora non considerato delle comunicazioni underwater: la dipendenza delle prestazioni di una rete di nodi in presenza al variare delle condizioni ambientali durante l'anno e al cambiare delle stagioni. È intuitivo che durante l'anno cambi la temperatura dell'acqua e quindi cambi di conseguenza anche l'SSP, fattore di grande impatto nella propagazione delle onde sonore.

Le Figure (3.37) e (3.38) presentano il throughput e il PER del protocollo

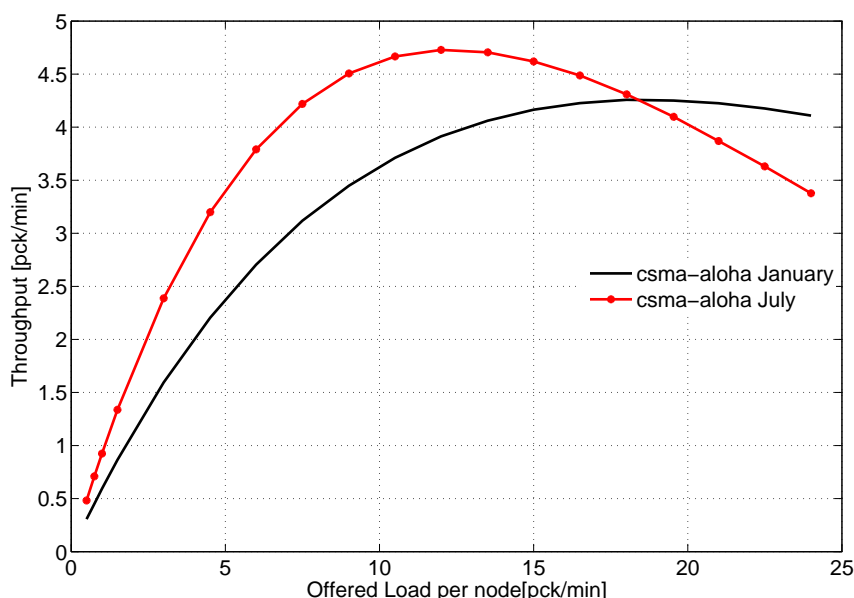


Figura 3.37: Confronto delle prestazioni del protocollo CSMA-ALOHA in due periodi diversi dell'anno. Throughput in funzione del traffico offerto

CSMA-ALOHA con i parametri del protocollo e del livello fisico precedenti, per due mesi diversi; uno estivo e uno invernale.

Le differenze, pur non essendo molto accentuate, sono evidenti nei due casi.

Essendo i nodi posizionati in maniera uguale nei due mesi ed essendo i parametri invariati, è evidente come la variazione delle prestazioni sia dovuta a un cambiamento delle condizioni del canale dovuta a una variazione degli SSP. Per meglio indagare questo fenomeno, però, non basta riferirsi ai grafici (3.2) e (3.3). Si ritiene utile, invece, andare a indagare l'SNR medio di ricezione dei pacchetti al sink in funzione della posizione del nodo, in assenza di interferenza. La Figura (3.39) riporta un istogramma degli SNR medi di ricezione di un pacchetto da parte del sink in funzione della posizione del nodo (partendo nel conteggio dal

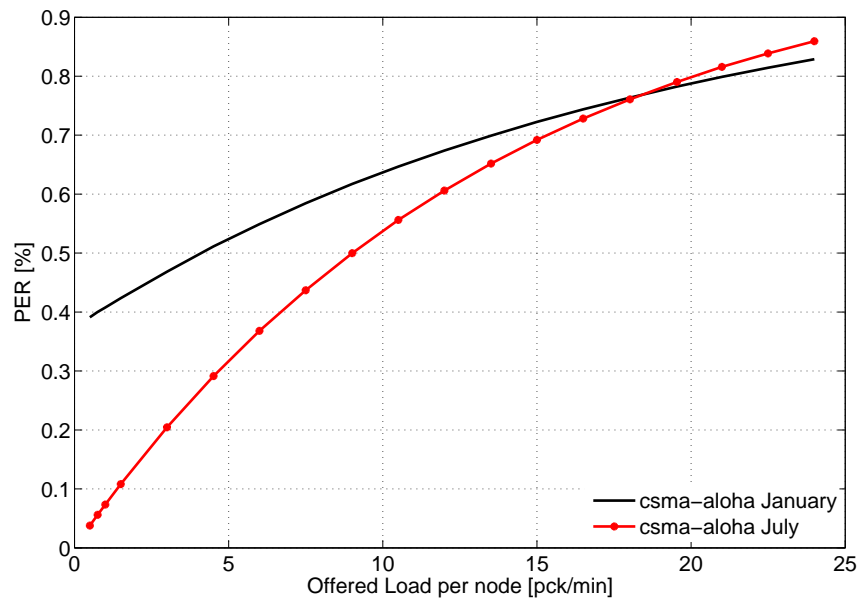


Figura 3.38: Confronto delle prestazioni del protocollo CSMA-ALOHA in due periodi diversi dell'anno. PER in funzione del traffico offerto

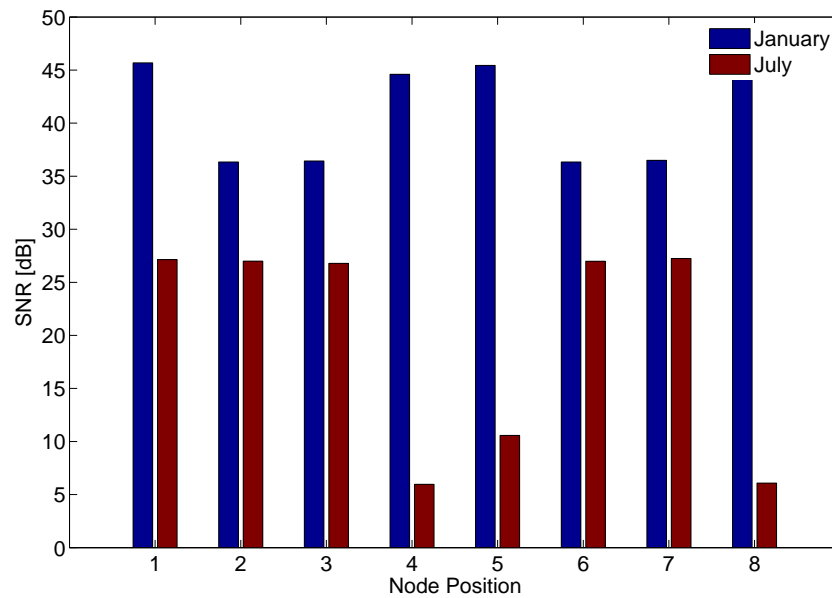


Figura 3.39: SNR [dB] medio con cui un pacchetto viene ricevuto al sink in funzione della stagione e della posizione del nodo.

nodo in basso a sinistra, continuando da sinistra verso destra nella stessa riga e poi sempre da sinistra verso destra nella riga superiore) e in funzione della sta-

gione dell'anno.

E' visibile come l'SNR sia costantemente superiore nel mese più freddo dell'anno e come tutti i nodi abbiano un largo margine di SNR rispetto alla soglia di decodificabilità. È altresì evidente come nel caso estivo alcuni nodi (in particolare il numero 4, 5 e 8) abbiano degli SNR di ricezione molto vicini alla soglia di decodificabilità, a causa di un canale particolarmente sfavorevole.

Questi nodi, in presenza di interferenza, quindi, avranno un throughput molto basso perchè di sovente i pacchetti da loro mandati arriveranno corrotti al sink. L'istogramma presentato quindi spiega chiaramente l'andamento del throughput e del PER dei grafici precedenti.

Si può notare, tuttavia, che, per traffici elevati in rete, si ha una inversione di tendenza e la rete nel caso estivo comincia ad avere delle performance migliori rispetto al caso invernale. Anche questo comportamento è spiegabile attraverso l'istogramma di Figura (3.39): in una rete con molto traffico come quella che stiamo esaminando in questo momento, il problema maggiore è rappresentato non dal rumore e dal path loss, ma bensì dall'interferenza mutua tra i nodi.

In questo senso, una maggiore attenuazione del canale aiuta a mitigare l'interferenza tra nodi sufficientemente distanti e quindi aiuta a disaccoppiare nello spazio le trasmissioni, portando a un miglioramento delle metriche in esame.

Le Figure (3.40) e (3.41) mostrano le prestazioni del protocollo USR-AIMD confrontato con il protocollo CSMA-ALOHA. Anche nel caso di ambiente deep-water valgono praticamente le stesse considerazioni fatte per il caso shallow-water, in cui possiamo notare effettivamente, ancora una volta, come il protocollo USR-AIMD non riesca ad avere prestazioni buone come nel caso dei nodi posizionati in maniera random. In particolare, come fatto precedentemente, si può concludere che l'interferenza centralizzata al sink costituisce un punto debole per il protocollo di ARQ. Tuttavia, il sistema di ACK e di ARQ permette al throughput di saturare a un valore medio senza diminuire. Per quanto riguarda il PER, valgono ancora una volta le considerazioni fatte nel caso shallow-water.

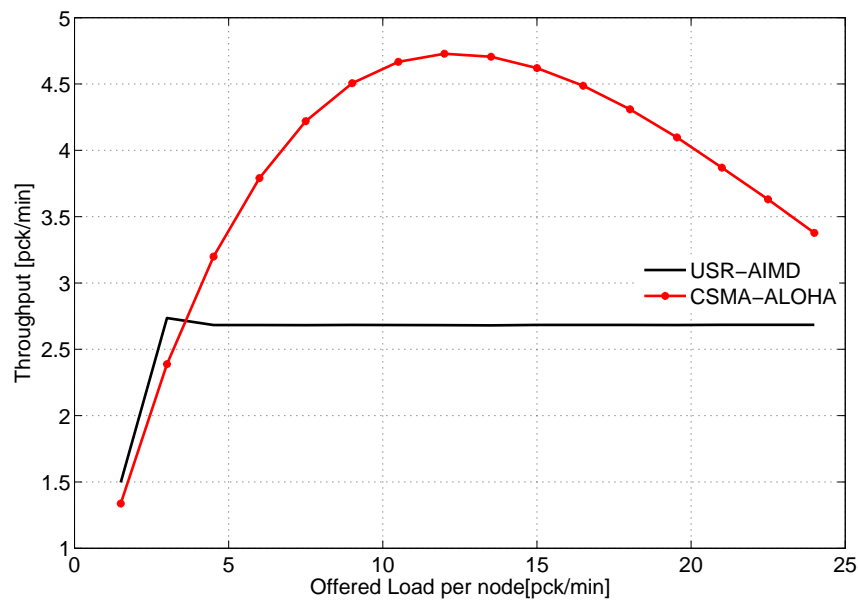


Figura 3.40: Throughput per nodo dei protocolli CSMA-ALOHA e USR-AIMD in funzione del traffico offerto per nodo.

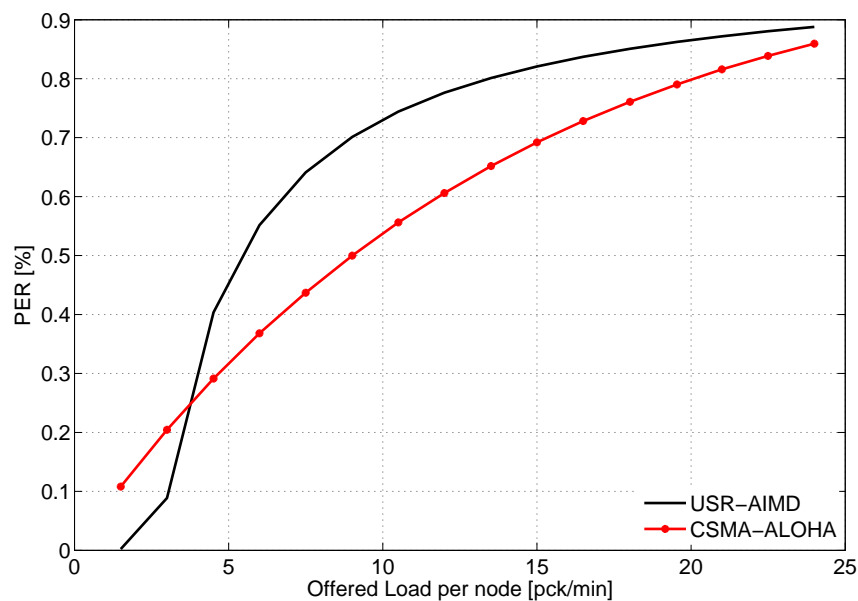


Figura 3.41: Packet Error Rate per nodo dei protocolli CSMA-ALOHA e USR-AIMD in funzione del traffico offerto per nodo.

3.3 Conclusioni e commenti

In conclusione, si possono fare alcuni commenti rispetto ai risultati ottenuti. In particolare, si può affermare che una politica di accesso casuale al canale, affiancata ad un breve tempo di sensing del canale, presenta delle prestazioni molto buone in tutti i casi presi in esame in questa tesi.

D'altra parte, i protocolli richiedenti un certo coordinamento tra i nodi come TONE-LOHI e DACAP sono limitati soprattutto dall'interferenza diffusa nella rete con disposizione omogenea e casuale dei nodi. In particolare, per il DACAP, è stato dimostrato che aumentando la potenza non migliorano necessariamente le prestazioni e, anzi, mantenendo una bassa potenza di trasmissione ma che permetta ugualmente di avere un SNR sufficiente al destinatario, l'interferenza creata nella rete è minore e le prestazioni migliorano. Per il TONE-LOHI si possono fare analoghi ragionamenti.

Nel caso di una disposizione random dei nodi e in cui ogni nodo comunica con tutti i suoi vicini, affiancare al CSMA-ALOHA una politica di ARQ SR con un calcolo adattivo del numero di pacchetti che è possibile trasmettere in un RTT porta a delle prestazioni in termini di throughput migliori. Il tasso di errore sul pacchetto aumenta ma, con un miglior utilizzo del canale, riesco a recapitare un maggior numero di pacchetti per unità di tempo al destinatario.

Nel caso di nodi che devo comunicare a un unico destinatario, invece, i protocolli come DACAP e TONE-LOHI permettono di migliorare il throughput e il PER a basso traffico, ma complessivamente il CSMA-ALOHA garantisce delle prestazioni complessivamente delle prestazioni migliori in qualsiasi condizione di traffico.

In questo caso, affiancare una politica di ARQ a quest'ultimo, invece, non porta a miglioramenti significativi e, anzi, porta a una saturazione del throughput a un valore più basso di quello raggiungibile da CSMA-ALOHA e a un peggioramento del Packet Error Rate.

Capitolo 4

UW-POLLING

Nei precedenti capitoli si sono esaminate le prestazioni dei protocolli di accesso al mezzo che sono stati presi in considerazione e si è evidenziato il fatto che il CSMA-ALOHA esibisce delle prestazioni complessivamente migliori in entrambi gli scenari proposti (deep water e shallow water).

In questo capitolo, viene introdotto un nuovo tipo di protocollo specifico per alcune applicazioni di comunicazione sottomarina.

Supponiamo di avere, in una certa zona del mare, una rete di sensori che monitorano un certo parametro significativo per qualche applicazione (per esempio presenza di plancton, fughe di gas in gasdotti sottomarini o altro).

Periodicamente, un veicolo autonomo (*AUV - Autonomous Underwater Vehicle*) mandato da una nave “pattuglierà” la rete con una traiettoria ben definita per “scaricare” i dati dai nodi e, tornando alla nave, aggiornare i ricercatori sulle nuove rilevazioni. Quando l’AUV sarà nella zona della rete, i nodi dovranno accedere al canale per trasmettere i dati all’AUV che in quel momento sta passando in quella zona. La topologia della rete è in qualche modo simile all’ultima vista nel capitolo precedente con nodi che trasmettono a un sink, che però in quel caso era fisso. Alla luce dei risultati ottenuti nel capitolo precedente, si potrebbe applicare una politica di accesso random al canale come il CSMA-ALOHA. Tuttavia, se la rete è sparsa, la scelta di un protocollo ad accesso completamente casuale, condurrebbe ad eccessive perdite di pacchetti.

Pertanto, in questo capitolo, presenteremo un protocollo di *polling*, in cui è proprio l’AUV a informare i nodi della sua presenza. Una volta che i nodi sono a conoscenza della presenza dell’AUV nel raggio di copertura, essi possono “preno-

4. UW-POLLING

tare” il canale mandando un messaggio di PROBE con il quale inviano all’AUV alcune informazioni utili per stilare un ordine di accesso, secondo una politica che dipende dall’applicazione. Una volta stilata questa classifica, l’AUV manderà un messaggio di POLL e il nodo prescelto avrà il canale riservato per sè per il tempo necessario per trasmettere una quantità di pacchetti precedentemente concordata con l’AUV tramite il messaggio di PROBE.

In questa maniera, si evita l’interferenza durante la trasmissione dei dati, a scapito dell’aumento dei tempi di attesa del protocollo rispetto a CSMA-ALOHA.

L’unica condivisione del canale avviene appunto quando i nodi dovranno trasmettere il PROBE per informare l’AUV della loro volontà di trasmettere i dati. In questo caso si adotta una politica di accesso random con backoff fisso.

Dopo aver visto in dettaglio il funzionamento del protocollo e le rispettive macchine a stati per il nodo e per l’AUV, si va a descrivere lo scenario di simulazione e a confrontare questo protocollo con gli altri protocolli presi in considerazione.

4.1 Il protocollo UW-POLLING (Nodo)

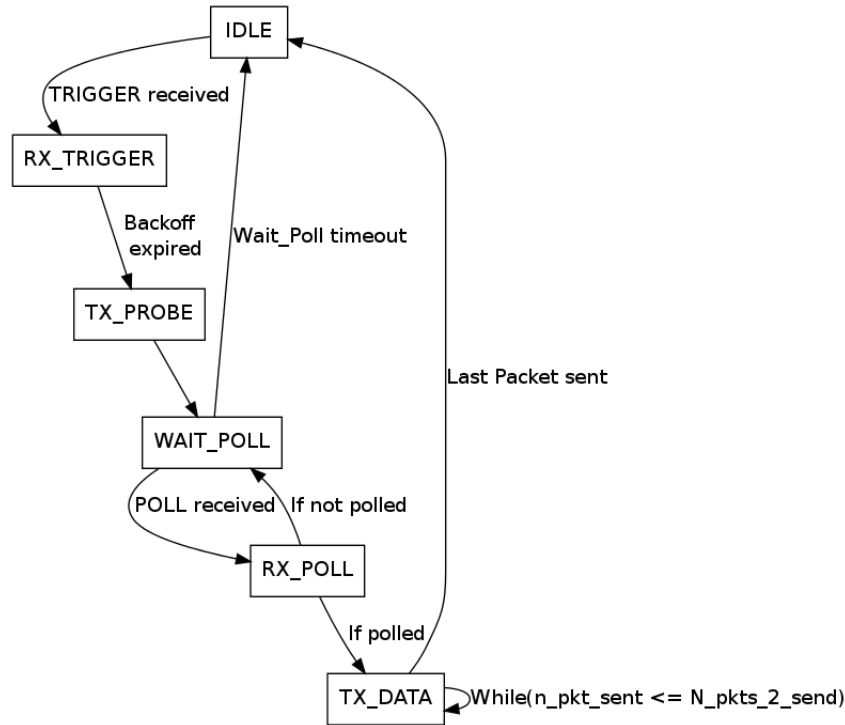


Figura 4.1: Macchina a stati per il protocollo UW-POLLING (nodo).

In Figura (4.1) è presentata la macchina a stati per il protocollo UW-POLLING lato nodo. Supponiamo che il nodo stia immagazzinando dati forniti da alcuni sensori montati. Esso parte da uno stato IDLE in cui l'unica attività presente è il sensing. Alla ricezione corretta di un messaggio di TRIGGER mandato in broadcast dall'AUV, il nodo passa in uno stato in cui imposta un tempo casuale di Backoff scelto tra un valore minimo e massimo mandati dall'AUV all'interno del messaggio di TRIGGER. Scaduto questo tempo di backoff, il nodo manda all'AUV un messaggio di PROBE in cui sono indicati il tempo di backoff scelto dal nodo stesso (utile all'AUV per calcolare il Round Trip Time), il timestamp del pacchetto dati più recente generato e il numero di pacchetti che avrà intenzione di mandare qualora sia "pollato".

La politica sulla base della quale l'AUV sceglierà chi trasmetterà per primo sarà basata sul timestamp più recente, in maniera da ricavare dalla rete le informazioni più recenti. Mandato il PROBE, il nodo imposta un tempo di attesa del POLL. Se, scaduto questo tempo, non riceve nessun messaggio di POLL, ritonerà nello

stato IDLE. Viceversa, se riceve un messaggio di POLL, andrà a verificare se è effettivamente lui il nodo “prescelto” per la trasmissione. Se non lo è, ritornerà nello stato di attesa del suo turno. Se, invece, è proprio lui il nodo prescelto per la trasmissione, manderà in ordine all’AUV il numero di pacchetti preventivamente comunicato. Alla fine di questa trasmissione, il nodo tornerà nello stato IDLE in attesa di un successivo messaggio di TRIGGER. Qualora il nodo ricevesse un TRIGGER ma non avesse nessun dato da trasmettere, semplicemente scarterà il TRIGGER e non risponderà all’AUV.

4.2 Il protocollo UW-POLLING (AUV)

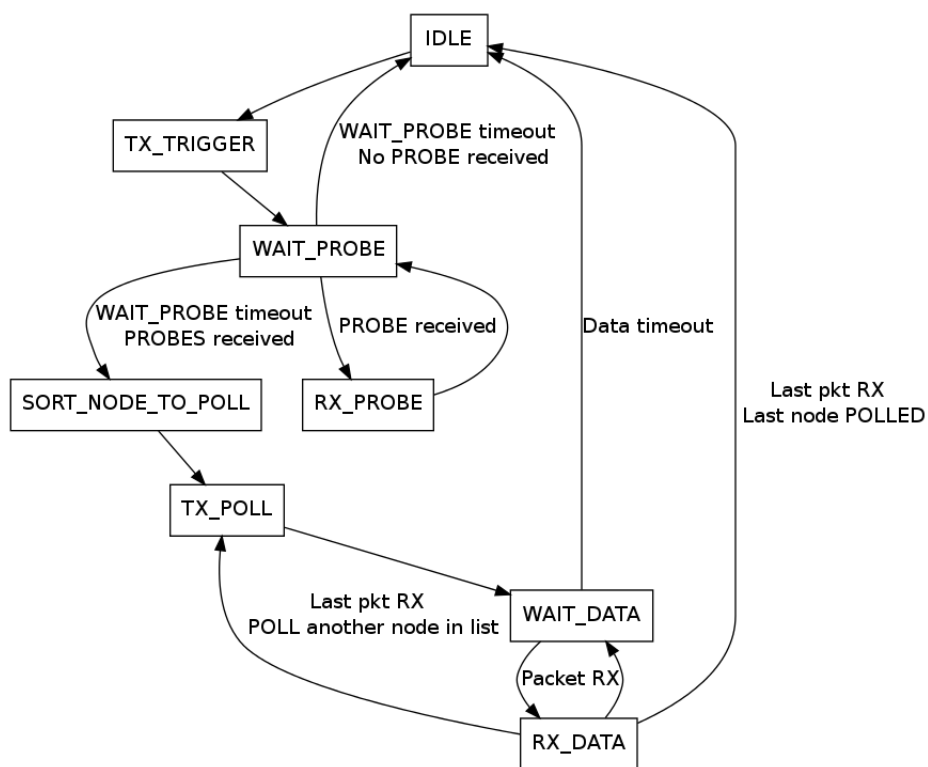


Figura 4.2: Macchina a stati per il protocollo UW-POLLING (AUV).

In Figura (4.2) è presentata la macchina a stati per l’AUV. Supponendo che l’AUV abbia raggiunto la rete, manderà un messaggio di TRIGGER e imposterà un tempo di attesa dei vari PROBE. A ogni PROBE ricevuto l’AUV (nello stato *RX_PROBE*) memorizzerà in una struttura dati interna le informazioni di ogni nodo che ha manifestato l’inten-

zione di trasmettere dati. Allo scadere del timer di ricezione dei PROBE, l'AUV verificherà il numero di PROBE effettivamente ricevuti. Nel caso nessun PROBE sia stato ricevuto, si può supporre di non aver nessun nodo in copertura oppure che nessun nodo abbia informazioni da trasmettere.

Nel caso contrario in cui ci siano dei nodi che hanno risposto correttamente tramite dei PROBE, l'AUV provvederà all'ordinamento dei nodi sulla base del pacchetto più recente generato. Effettuato l'ordinamento, l'identificativo dei nodi (chiamato *id_node* nel messaggio di POLL) vengono inseriti in ordine dentro in un array dentro il messaggio di POLL e quest'ultimo viene spedito in broadcast nella rete. Ogni nodo verificherà quindi se è il primo della lista e, in caso contrario, ignorerà il messaggio. Si è scelto di trasmettere l'insieme completo dei nodi che saranno "POLLATI" perchè si prevede, in futuro, di sviluppare una versione in cui si fa sapere al nodo che è in lista quanto dovrà aspettare prima che sia il suo turno, in maniera che il nodo possa andare in uno stato di risparmio energetico e possa "svegliarsi" poco prima che sia il suo turno. Trasmesso il POLL, l'AUV imposta un tempo di time-out proporzionale al RTT tra sè e il nodo prescelto per il polling e al numero di pacchetti che si aspetta dal suddetto nodo. Terminata questa fase con lo scadere del timer o la ricezione corretta di tutti i nodi, l'AUV passa a "pollare" il nodo successivo in lista oppure a mandare un altro TRIGGER, nel caso in cui tutti i nodi in lista siano stati selezionati e abbiano avuto la possibilità di trasmettere i propri dati.

Per esigenze di spazio e di complessità computazionale, si è deciso di fissare un numero massimo di nodi con cui comunicare e un numero massimo di pacchetti accettabili da un nodo. La prima scelta è dovuta al fatto che se troppi nodi si "prenotano" per la trasmissione, visto che l'AUV è in movimento, c'è il rischio che l'ultimo nodo della lista trasmette quando l'AUV si è già spostato oltre la portata della trasmissione o ha un canale poco affidabile. La seconda scelta, invece, è stata adottata per evitare che un nodo che ha tante informazioni occupi per troppo tempo il canale mandando troppi pacchetti all'AUV.

4.3 Scenario di simulazione e risultati

4.3.1 Zona di simulazione e posizionamento dei nodi

La simulazione avviene nel mare Adriatico nella zona della regione Puglia, in Gennaio, quindi con SSP invernale (Figura (4.3)). In Figura 4.4 è presentato anche il path loss del canale ottenuto con Bellhop per la zona prescelta.

Le coordinate associate alla zona prescelta sono

- **Latitudine:** 41.90 °N
- **Longitudine:** 17.51 °E

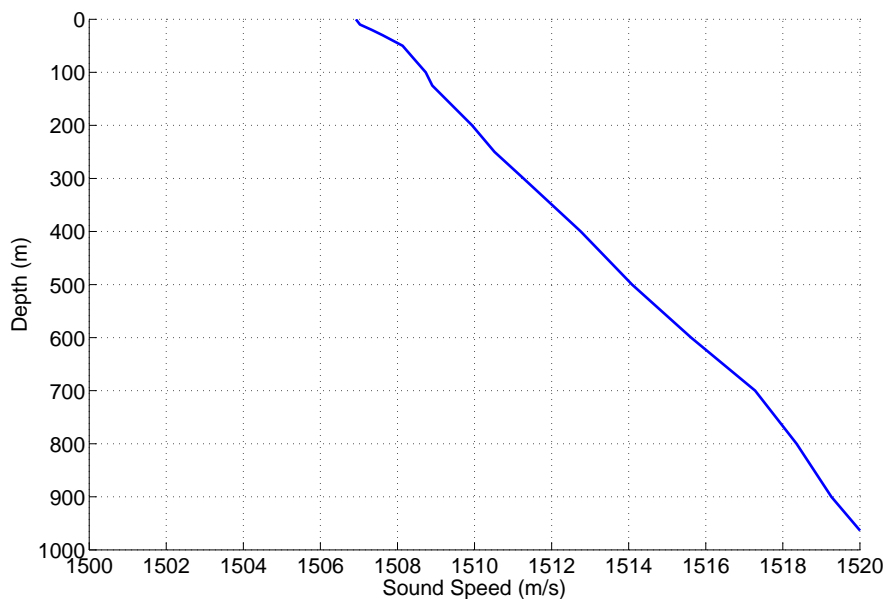


Figura 4.3: Sound Speed Profile del Mar Adriatico alle coordinate (**41.90° N**, **17.51° E**).

In questa zona sono stati posizionati sul fondale (circa 1000 metri di profondità) 25 nodi a griglia con 5 nodi ogni riga, distanti tra di loro 2 Km.

L'AUV sorvola la rete a una profondità di 600 m con una velocità di 4 nodi (circa 2 m/s) seguendo la traiettoria in Figura (4.5) eseguendo quattro sorvoli completi durante la simulazione.

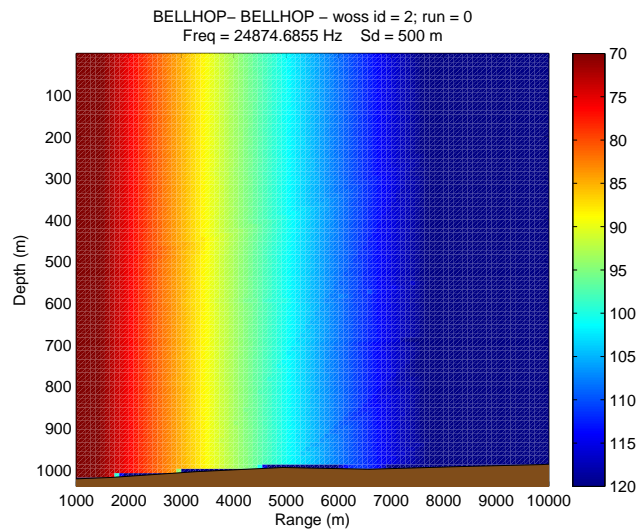


Figura 4.4: Path loss riferito all'SSP di Figura (4.3) calcolato in maniera incoerente.

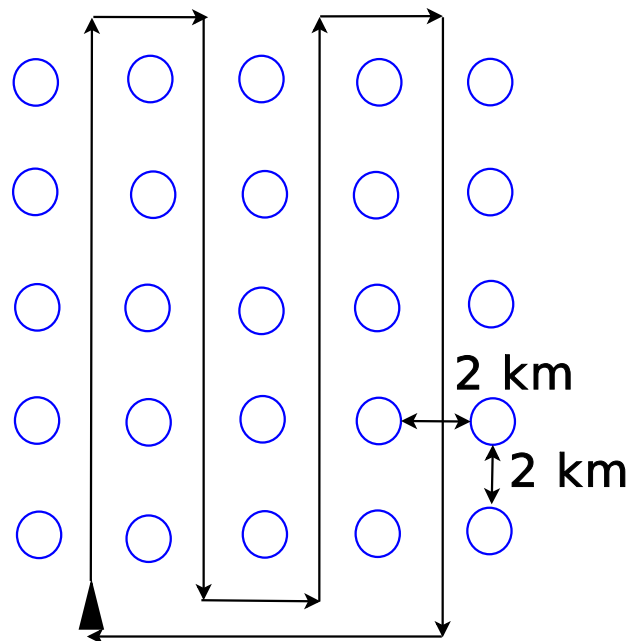


Figura 4.5: Topologia della rete e traiettoria di sorvolo dell'AUV.

4.3.2 Risultati delle simulazioni

Il protocollo UW-POLLING viene confrontato con i protocolli CSMA-ALOHA e DACAP con varie potenze trasmissive e livelli di traffico generato dai nodi.

In questo modo, si confronteranno, oltre che i protocolli in sè, anche varie politiche di trasmissione in uno scenario come quello appena descritto.

4. UW-POLLING

In particolare, si confrontano due protocolli con due livelli di coordinazione tra i nodi (nel caso del protocollo di polling i nodi sono completamente coordinati dall'AUV che decide l'ordine di trasmissione, mentre nel caso del DACAP i nodi contendono l'accesso al canale tramite l'invio di un RTS) con un protocollo dove è assente la coordinazione e l'accesso al canale avviene in maniera casuale.

I parametri scelti per le simulazioni sono uguali a quelli scelti nel capitolo precedente:

- **Dimensione dei pacchetti dati:** 125 byte
- **Modulazione utilizzata:** BPSK
- **Frequenza della portante:** 25 kHz o 11.5 kHz
- **Bitrate:** 4800 bps
- **Banda:** 5 kHz
- **AcquisitionThreshold:** 5 dB

I time-out di ricezione del PROBE da parte dell'AUV e del POLL da parte del nodo sono stati calcolati in modo che l'AUV comunichi al massimo con 10 nodi circostanti. Si suppone che il nodo più lontano disti 3.5 Km dall'AUV e ogni nodo possa trasmettere all'AUV un massimo di 20 pacchetti per ogni turno.

Assumendo quindi una propagazione dei segnali acustici a una velocità di 1500 m/s e data la topologia della rete, i valori di timer sono rispettivamente 10 secondi per la ricezione del PROBE e 15 secondi per la ricezione del POLL. Nelle tre sezioni seguenti verrà variata la potenza trasmissiva, osservando le prestazioni dei protocolli per i tre valori considerati e per alcune scelte del periodo di tempo che intercorre tra la generazione di due pacchetti consecutivi da parte del modulo CBR (brevemente chiamato "periodo CBR"). Verranno di volta in volta commentati i risultati e alla fine verranno tratte le conclusioni.

Potenza trasmissiva: 130 dB re μPa .

Nelle tabelle (4.1) (4.2) e (4.3) sono presentati i risultati dei tre protocolli esaminati, in termini di throughput, Packet Error Rate (PER) e numero di pacchetti complessivamente consegnati all'AUV. Osservando i valori nelle tabelle si nota che, ad alto traffico, il protocollo UW-POLLING riesce effettivamente ad ottenere

UW-POLLING			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	32.2	13	¿ 57k
150	21.04	6	¿ 37k
30	5.18	17	¿ 9k
15	3	24	¿ 5k

Tabella 4.1: Risultati per il protocollo UW-POLLING per vari livelli di traffico

CSMA-ALOHA			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	34.75	89	¿ 58k
150	19.73	87	¿ 33k
30	4.43	86	¿ 7k
15	2.23	86	¿ 4k

Tabella 4.2: Risultati per il protocollo CSMA-ALOHA per vari livelli di traffico

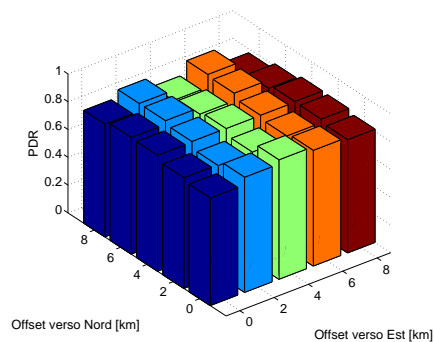
DACAP			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	10	3	18k
150	10.8	3	> 18k
30	10.3	3	> 17k
15	4	6	> 6k

Tabella 4.3: Risultati per il protocollo DACAP per vari livelli di traffico

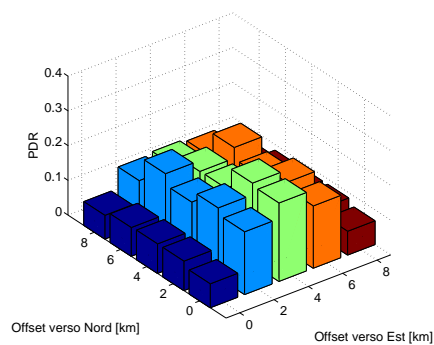
throughput comparabili con il CSMA-ALOHA, con PER più bassi. I pacchetti persi con UW-POLLING sono dovuti sostanzialmente a trasmissioni che avvengono quando l'AUV si è allontanato e il canale è inaffidabile, mentre, per quanto riguarda CSMA-ALOHA, oltre ai pacchetti persi per SNR insufficiente, i pacchetti sono soggetti a collisioni all'AUV se trasmessi simultaneamente o quasi (evento probabile, specialmente per periodi CBR ridotti). DACAP, tuttavia, riesce a trasmettere i pacchetti in maniera molto affidabile (Packet Error Rate minore) ma riesce a trasmetterne pochi (ottenendo così un basso throughput) a causa dei continui eventi di *defer* causati dalla ricezione di RTS e CTS multipli da parte dei nodi e dell'AUV. Tuttavia, abbassando il traffico, DACAP esibisce un throughput

leggermente superiore a UW-POLLING. Questo fatto è dovuto al minor tempo di cui necessita il DACAP per trasmettere un pacchetto.

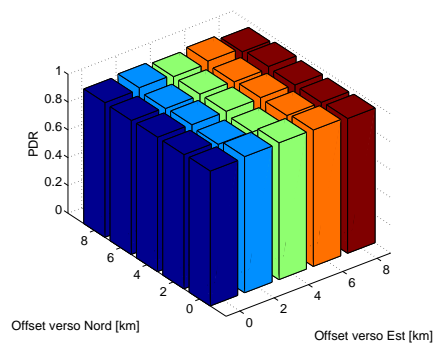
In Figura (4.6) è mostrato, inoltre, il PDR (Packet Delivery Ratio) per ogni nodo, in funzione alla posizione dei nodi. Con CSMA-ALOHA, vengono privilegiati i nodi centrali della rete. Infatti, usando CSMA-ALOHA, i nodi trasmettono i dati indipendentemente dalla presenza dell'AUV. Quindi, i nodi che hanno verso l'AUV un canale migliore per un tempo più lungo, riescono a recapitare più pacchetti rispetto agli altri. I due protocolli UW-POLLING e DACAP, invece, danno a tutti i nodi la stessa possibilità di trasmettere i dati perchè è l'AUV che informa i nodi della propria presenza, autorizzandoli a trasmettere i dati (tramite un TRIGGER o un CTS rispettivamente).



(a) UW-POLLING



(b) CSMA-ALOHA



(c) DACAP

Figura 4.6: PDR (Packet Delivery Ratio) in funzione della posizione dei nodi per i tre protocolli. Il periodo di generazione di pacchetti è fissato a 50 s.

Potenza trasmittiva: 150 dB re μPa .

In questa sezione viene aumentata la potenza trasmittiva fino al valore di 150 dB re μPa . CSMA-ALOHA presenta delle prestazioni migliori rispetto al caso precedente. I nodi hanno un canale migliore verso l'AUV per più tempo, quindi

4. UW-POLLING

riescono a trasmettere pacchetti al SINK per più tempo, ottenendo un throughput complessivamente migliore. UW-POLLING presenta anch'esso delle prestazioni migliori sia a basso che ad alto traffico.

Tuttavia, il PER di CSMA-ALOHA risulta ancora maggiore rispetto a UW-POLLING a causa delle collisioni. Inoltre, a basso traffico, la coordinazione richiesta da UW-POLLING permette di ottenere prestazioni migliori. DACAP invece, come già notato nel capitolo precedente, viene penalizzato dall'aumento dell'interferenza dovuto all'aumento della potenza. Infatti, il throughput risultante è il più tra i tre protocolli, e il Packet Error Rate, pur essendo ancora il migliore a traffico alto, diventa maggiore di quello di UW-POLLING con la rete scarica. La potenza elevata che permette di raggiungere l'AUV più lontano e il numero di pacchetti limitato che ogni nodo deve trasmettere, permette a UW-POLLING di essere molto efficiente a basso traffico, ottenendo un throughput comparabile a quello del CSMA-ALOHA con un Packet Error Rate migliore. I dati sono riportati nelle Tabelle (4.4), (4.5) e (4.6). In Figura 4.7 sono visibili invece i PDR dei vari nodi con un traffico fissato a un pacchetto ogni 50 secondi. Ancora una volta è visibile come, sia DACAP che UW-POLLING permettono a tutti i nodi di trasmettere correttamente pacchetti mentre CSMA-ALOHA privilegia i nodi centrali nella rete.

UW-POLLING			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	33.67	22	60k
150	34.69	17	< 62k
30	15	5	< 27k
15	8.25	1	< 15k

Tabella 4.4: Risultati per il protocollo UW-POLLING per vari livelli di traffico

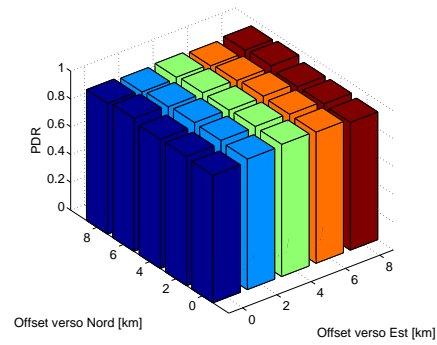
CSMA-ALOHA			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	67.96	77	> 118k
150	46.48	69	81k
30	13	57	> 22k
15	6.8	55	< 12k

Tabella 4.5: Risultati per il protocollo CSMA-ALOHA per vari livelli di traffico

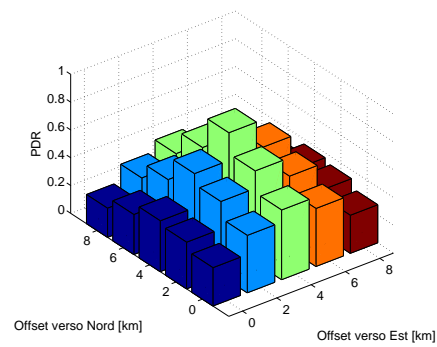
DACAP			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	5.75	8	< 10k
150	5.73	8	< 10k
30	5.72	8	< 10k
15	4.46	7	> 7k

Tabella 4.6: Risultati per il protocollo DACAP per vari livelli di traffico

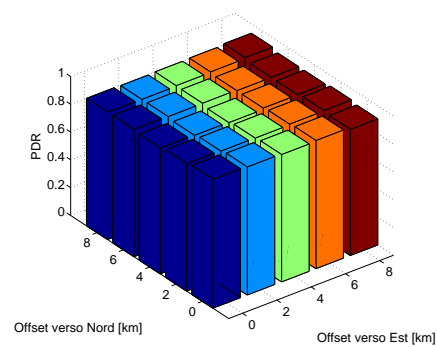
4. UW-POLLING



(a) UW-POLLING



(b) CSMA-ALOHA



(c) DACAP

Figura 4.7: PDR (Packet Delivery Ratio) in funzione della posizione dei nodi per i tre protocolli. Il periodo di generazione di pacchetti è fissato a 50 s.

Potenza trasmisiva: 190 dB re μPa .

Nei risultati che seguono viene alzata ulteriormente la potenza trasmisiva al valore di 190 dB re μPa

UW-POLLING			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	30	23	> 53k
150	30	21	> 54k
30	4	5	> 37k
15	7	1	> 21k

Tabella 4.7: Risultati per il protocollo UW-POLLING per vari livelli di traffico

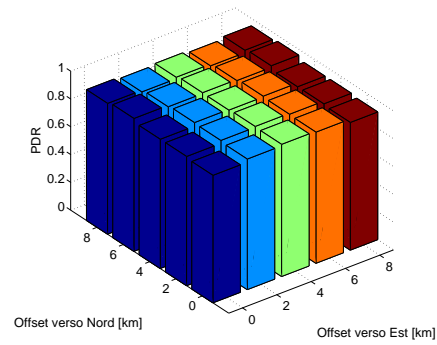
CSMA-ALOHA			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	91.38	69	< 163k
150	70.7	52	126k
30	24.67	17	< 44k
15	13.56	9	> 24k

Tabella 4.8: Risultati per il protocollo CSMA-ALOHA per vari livelli di traffico

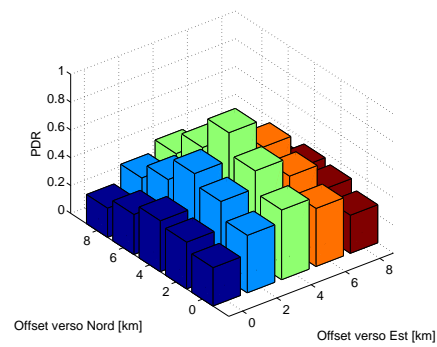
DACAP			
Traffico [pkt/min]	Throughput [pkt/min]	PER [%]	PKT consegnati al sink
300	5.75	8	< 10k
150	5.73	8	< 10k
30	5.72	8	> 9k
15	4.46	7	> 7k

Tabella 4.9: Risultati per il protocollo DACAP per vari livelli di traffico

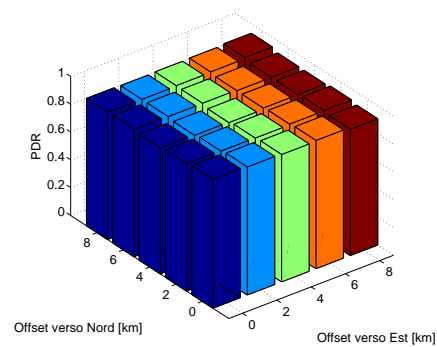
4. UW-POLLING



(a) UW-POLLING



(b) CSMA-ALOHA



(c) DACAP

Figura 4.8: PDR (Packet Delivery Ratio) in funzione della posizione dei nodi per i tre protocolli. Il periodo di generazione di pacchetti è fissato a 50 s.

In questo caso le prestazioni del protocollo UW-POLLING e del protocollo DACAP risultano praticamente invariate rispetto al caso precedente. Il protocollo DACAP, come prima, esibisce un basso throughput a causa dell'interferenza degli RTS e CTS provenienti dai vari nodi. In questo caso, inoltre, l'aumento della potenza permette ai messaggi di controllo di propagarsi a distanze maggiori, rendendo più probabile un *defer* di una trasmissione e quindi il dispendio di tempo non utilizzato per trasmissioni di pacchetti utili.

CSMA-ALOHA esibisce prestazioni migliori rispetto al caso precedente. L'elevata potenza trasmessa permette anche i nodi più distanti di comunicare con l'AUV. Anche il PER è migliore rispetto ai casi precedenti perchè, essendo la potenza maggiore i nodi hanno un canale migliore per un tempo più lungo e, inoltre, il carrier sensing è in grado di sentire trasmissioni provenienti da più lontano, rendendo meno probabili le collisioni.

Tuttavia, il PER del CSMA-ALOHA è maggiore degli altri due protocolli considerati.

Per i PDR dei nodi (Figura 4.8) possono essere tratte le stesse conclusioni.

4.4 Conclusioni e commenti

In questo capitolo è stato presentato un protocollo di polling e ne sono state analizzate le prestazioni in una rete con AUV in movimento, confrontandole con quelle di altri protocolli studiati precedentemente.

Le applicazioni tipiche di una rete così progettata sono il monitoraggio di vari parametri ambientali di interesse in scenari oceanici difficilmente raggiungibili con comunicazioni completamente wireless o usando connessioni dati via cavo.

È stato evidenziato come CSMA-ALOHA, per potenze elevate, presenti delle prestazioni migliori in termini di throughput rispetto al protocollo di polling studiato. Per l'applicazione in esame, però, CSMA-ALOHA richiede un dispendio di energia maggiore, visto che trasmette pacchetti senza sapere se l'AUV si trova entro il raggio di copertura della trasmissione o no. Questo è il motivo, assieme alle collisioni, per cui CSMA-ALOHA presenta un PER superiore degli altri due protocolli. Con UW-POLLING, l'AUV riesce a ricevere la maggior parte dei dati inviati dai nodi anche a potenze basse, ottenendo prestazioni leggermente migliori rispetto a CSMA-ALOHA. Quindi, per quanto appena detto, il protocollo UW-POLLING può essere impiegato con buoni risultati tenendo conto anche dell'efficienza energetica e del risparmio della potenza del nodo.

Utilizzando UW-POLLING, tuttavia, i nodi creano una sorta di buffer a livello MAC per mantenere in memoria i dati generati, non potendo trasmetterli appena sono pronti come invece avviene con CSMA-ALOHA. Si rende necessario, quindi, impostare un numero massimo di pacchetti trasmissibili per ogni nodo. Sceglierne un quantitativo limitato, penalizza il throughput a traffico alto, perchè non tutti i pacchetti generati trovano posto nella trasmissione. D'altra parte, aumentare il numero di pacchetti trasmissibili per ogni nodo, porta il nodo stesso a occupare il canale per più tempo. Così facendo si allungano i tempi di attesa per tutti gli altri nodi in lista ed aumenta il rischio che l'AUV sia allontanato troppo quando l'ultimo nodo in una lista di polling deve trasmettere.

Infine, DACAP, pur esibendo PER molto buoni, presenta un throughput basso a causa dell'interferenza di altri nodi che provocano l'annullamento delle trasmissioni dei dati utili.

Dal punto di vista dell'applicazione, questo problema può essere limitante. Si suppone, infatti, che l'AUV che viene mandato periodicamente a "scaricare" i dati fino ad allora memorizzati dai vari nodi, desideri ricevere il maggior numero

di pacchetti possibile. Da questo punto di vista, il protocollo UW-POLLING ha delle prestazioni che si accordano meglio con il risultato voluto.

Se il risparmio energetico nella rete non rappresenta un vincolo di progetto, mantenere una potenza trasmittiva elevata e utilizzare CSMA-ALOHA, permette di raggiungere un throughput elevato, sebbene il PER sia piuttosto alto. È possibile, invece, raggiungere delle prestazioni buone e migliori di CSMA-ALOHA sia in termini di throughput che di PER, mantenendo una potenza trasmittiva bassa e utilizzando il protocollo di polling presentato in questo capitolo. Questa scelta permette sicuramente un risparmio energetico nei nodi, sia perchè trasmettono i loro dati solo in presenza dell'AUV, sia perchè la potenza di trasmissione può essere mantenuta a livelli modesti. Una possibile alternativa è quella di modificare il protocollo CSMA-ALOHA, in maniera che sia l'AUV ad informare i nodi della sua presenza. In questo modo, solo dopo aver ricevuto correttamente questo avviso, i nodi possono contendere tramite carrier sensing e trasmettere i loro dati per un certo periodo di tempo.

Appendice A

Bellhop ray-tracing

Bellhop [13] è un programma per la risoluzione delle equazioni di propagazione delle onde acustiche in ambiente sottomarino scritto da Michael Porter e inserito nel pacchetto *Acoustic Toolbox* disponibile al sito

<http://oalib.hlsresearch.com/Rays>. Bellhop utilizza a questo fine un'approssimazione a raggi per onde acustiche in due dimensioni per un dato insieme di condizioni al contorno, fornite sotto forma di parametri ambientali. Il calcolo della pressione acustica è basato sulla teoria dei fasci Gaussiani [19] Bellhop si basa sulla risoluzione in due dimensioni delle equazioni *dynamic ray equations* che in coordinate cilindriche possono essere scritte come

$$\frac{dr}{ds} = c\xi(s) \quad \frac{d\xi}{ds} = -\frac{1}{c^2} \frac{\partial c}{\partial r} \quad (\text{A.1})$$

$$\frac{dz}{ds} = c\zeta(s) \quad \frac{d\zeta}{ds} = -\frac{1}{c^2} \frac{\partial c}{\partial z} \quad (\text{A.2})$$

dove $r(s)$ e $z(s)$ rappresentano le coordinate cilindriche del raggio, s la lunghezza dell'arco del raggio e la coppia $c(s)[\xi(s), \zeta(s)]$ sono le coordinate del versore lungo il raggio. Le condizioni iniziali per la risoluzione delle equazioni sono

$$r(0) = r_s \quad z(0) = z_s \quad (\text{A.3})$$

$$\xi(0) = \frac{\cos\theta_s}{c_s} \quad \zeta(0) = \frac{\sin\theta_s}{c_s} \quad (\text{A.4})$$

dove (r_s, z_s) sono le coordinate della sorgente e θ_s è l'angolo di lancio del raggio. In questo modo si può calcolare anche il tempo di arrivo del raggio tramite la formula

$$\tau = \int_{\Gamma} \frac{ds}{c(s)} \quad (\text{A.5})$$

dove Γ è la traiettoria del raggio in questione. L'algoritmo è scritto in Fortran e fornisce anche una serie di funzioni matlab per ottenere dei grafici utili alla comprensione del modello di canale.

Il modo di funzionamento di Bellhop preso in considerazione in questa tesi prevede la necessità di dare in input a Bellho due file, uno di formato *.bty* e il secondo di formato *.env*. Il primo dà delle informazioni sulla batimetria della zona scelta, mentre il secondo fornisce al programma di calcolo tutto il resto delle informazioni, come compisizione del fondo marino, l'SSP e il metodo di calcolo dell'attenuazione e altri parametri. Si descrive di seguito la struttura di questi file con un esempio pratico. All'inizio, prendiamo un file di batimetria.

```
'L'
10
      0          3973
    0.15        3974
      0.3        3974
    0.45        3973
      1.8        3973
    1.95        3967
    2.55        3967
      2.7        3963
    2.85        3963
      3          3958
```

Questo file è di semplice comprensione. La prima lettera in alto indica il metodo utilizzato per interpolare i punti di batimetria ottenuti ¹. Successivamente si indica il numero di punti presenti e poi, nella tabella sottostante, si indica la posizione (in km) e la profondità fondale marino in quel punto (in m).

Il file *.env* è più complicato ed è composto da varie voci

¹Le due possibilità in questo caso sono 'L' che sta per lineare, o 'C' che indica una interpolazione cubica

A. BELLHOP RAY-TRACING

TITLE

Frequency (in Hz)

nmedia (dummy integer < 20)

OPTIONS1

SURFACE-LINE

nmesh sigmas $z(nssp)$ |

$z(1)$ $cp(1)$ / |

$z(2)$ $cp(2)$ / | Sound

. . | Speed

. . | Block

. . |

$z(nssp)$ $cp(nssp)$ / -|-

OPTIONS2 $sigmab$ | Bottom

BOTTOM-LINE -|- Block

nsources (number of sources) |

source-depth(1) source-depth(nsources) / (in m) |

nrd (number of receivers x depth) | Array

4receiver-depth(1) receiver-depth(nrd) / (in m) | Block

nrr (number of receivers x range) |

receiver-range(1) receiver-range(nrr) / (in km) -|-

OPTIONS3 |

nbeams (number of launching angles) | Output

$\theta(1)$ $\theta(nbeams)$ (launching angles in degrees) | Block

ray-step $zmax$ $rmax$ -|-

OPTIONS4 $epmult$ $rloop$ $isingl$ | Beam

nimage $ibwin$ component -|- Bloc

Questo file sostanzialmente elenca le caratteristiche dell'ambiente in esame che hanno impatto sulla propagazione sonora. In particolare, possiamo notare l'indicazione della frequenza portante della trasmissione. Inoltre, si a fornire al programma l'SSP della zona in questione nella sezione indicata come *Sound Speed Block* nel formato [profondità (in m) velocità del suono (in m/s)].

Sotto a questi dati si possono indicare le coordinate della posizione della sorgente sonora e si possono definire le coordinate della griglia di punti dove sarà calcolato il valore della pressione acustica. Ovviamente, più fitta è questa griglia, più accurata è la rappresentazione degli effetti di propagazione. D'altra parte, la computazione dal punto di vista di tempo e risorse sarà più onerosa. Da notare anche il fatto molto importante che in `OPTIONS3` si può indicare il tipo di output desiderato. Le possibilità per questo parametro sono

- 'A' In questo caso viene generato un file **.arr* in cui vengono scritti i tempi di arrivo e le ampiezze di ciascun raggio e i tempi di arrivo per ciascun punto della griglia precedentemente definita
- 'C' In questo caso, invece, viene restituito un file *.shd* (che, se aperto con la funzione `plotshd` di Matlab restituisce una Figura come (1.9)) in cui il calcolo del path loss avviene in maniera coerente, cioè come il modulo della somma al quadrato delle ampiezze dei raggi.
- 'I' Anche in questo caso viene restituito lo stesso tipo di file di prima, con la sola differenza che in questo caso il path loss viene calcolato in maniera incoerente, cioè come somma dei moduli quadri delle ampiezze dei raggi
- 'S' In questo caso il file **.shd* viene generato tramite un calcolo semi-coerente.
- 'R' Restituisce invece le traiettorie dei raggi
- 'E' Evidenzia, invece, i raggi che giungono a una determinata destinazione decisa precedentemente.

Quindi, a seconda delle esigenze dell'utente, Bellhop è in grado di fornire varie informazioni riguardo alla propagazione dei raggi nell'ambiente sottomarino.

Infine, in questo file di input si possono indicare anche il numero di raggi che si

A. *BELLHOP RAY-TRACING*

desidera che Bellhop lanci prima di considerare conclusa la simulazione e l'angolo massimo e minimo di lancio del raggio. Si sottolinea il fatto che lasciando 0 come valore di raggi da lanciare dalla sorgente, si lascia a Bellhop la decisione del numero più consono di raggi da lanciare per raggiungere una sufficiente accuratezza, sulla base delle dimensioni dell'area e della distanza del ricevitore.

Per maggiori informazioni e per una spiegazione più dettagliata del formato del file di input si rimanda alla guida consultabile al sito

<http://oalib.hlsresearch.com/Rays/GeneralDescription.pdf>

Bellhop fornisce al simulatore, assieme a WOSS che sarà discusso nell'appendice successiva, la possibilità di avere un modello di canale più preciso rispetto al solo utilizzo di formule empiriche a spese di una maggiore complessità computazionale.

Appendice B

Ns-Miracle e WOSS

In questa appendice si va a descrivere le altre parti del simulatore utilizzato per questa tesi.

Ns-Miracle (Multi Interface Cross Layer Extension) è un'una estensione al programma Network Simulator 2 per simulazione di reti.

MIRACLE sostanzialmente fornisce la possibilità di simulare interfacce di comunicazione multiple (es. Bluetooth e WiFi) su uno stesso nodo. In aggiunta, fornisce metodi per implementare una comunicazione *cross-layer* tra i vari protocolli componenti lo stack protocollare di un dato nodo.

Inoltre, MIRACLE permette la gestione più flessibile e “libera” dei protocolli, permettendo di omettere per comodità protocolli superflui, oppure sostituendoli con protocolli più adatti. I principali protocolli, quindi, sono stati riscritti aggiungendo a ogni protocollo la possibilità di ricevere messaggi cross-layer sia dai livelli superiori che dai livelli inferiori. I vari protocolli sono interconnessi tramite entità chiamate Moduli. Ogni modulo è indipendente da ogni altro e può essere interconnesso con altri moduli per formare uno stack-protocollare. Esiste anche la possibilità che siano presenti più di un modulo per ogni livello protocollare inteso dal modello OSI. Ogni Modulo è collegato a un'altra entità chiamata *Node Core* che ha lo scopo di “ordinare” gerarchicamente ogni modulo in uno stack funzionante e, soprattutto, di coordinare lo scambio di messaggi tra moduli. Inoltre, ogni modulo è connesso con gli altri moduli soprastanti e sottostanti tramite dei SAP (Service Access Point) come prescrive anche il modello ISO/OSI.

Per esemplificare l'utilità di una estensione che fornisce una gestione completa della messaggistica cross-layer, si considerano i protocolli di livello MAC di cui

abbiamo parlato nel capitolo 1 e che sono stati implementati all'interno di Miracle. Grazie a Miracle, ogni protocollo MAC ha la possibilità di chiedere al livello sottostante (in questo caso il livello fisico) se la trasmissione in corso è terminata o meno, oppure ha la possibilità di ricevere la notifica della ricezione di un certo tipo di pacchetto. Tutte queste informazioni, permettono al protocollo di prendere delle decisioni tempestive sulle azioni da intraprendere.

Ogni livello del modello ISO/OSI è rappresentato in Miracle da una classe. Ogni sviluppatore che si accinge a costruire un nuovo protocollo di qualsiasi livello può ereditare la classe del suo progetto dalla classe già esistente, ereditando numerosi metodi generici utili per lo sviluppo.

Per quanto riguarda le comunicazioni underwater, Miracle mette a disposizione numerosi protocolli di livello MAC e un modello di canale basato sulle formule di Thorp e Urick [4]. L'esigenza della simulazione dei protocolli di livello superiore su un canale più realistico ha portato allo sviluppo del framework *WOSS* (World Ocean Simulation System). *WOSS* è un motore di simulazione inserito in *MIRACLE* e consente due possibilità: è possibile, nuovamente, utilizzare il modello di Thorp e Urick, più semplice dal punto di vista computazionale;

alternativamente si integra a *WOSS* l'utilizzo di Bellhop che fornisce le informazioni necessarie alla costruzione del modello di canale. Dal momento che Bellhop richiede l'utilizzo di alcune informazioni specifiche della zona in cui si desidera effettuare la simulazione, *WOSS* si appoggia ad alcuni database oceanografici [9] [10] e [11], ottenendo, tramite delle query, le informazioni necessarie a Bellhop. Grazie a questa automazione, l'utente finale deve soltanto inserire le coordinate geografiche della zona dove intende simulare la rete. *WOSS* si occuperà di interrogare i database e costruire di conseguenza il file `.env` che Bellhop richiede.

Bellhop, a sua volta, fornisce le informazioni a *WOSS* utili per ricostruire il canale nella zona specificata e *WOSS* elabora le informazioni appena ricevute per calcolare il power-delay profile da passare a NS-2 come modello di canale.

Questa metodologia permette, pagando dal punto di vista della complessità computazionale, di avere un affidabile modello di canale in qualsiasi zona del mondo, semplicemente scegliendo le coordinate geografiche.

WOSS permette inoltre di simulare la variazione delle condizioni ambientali durante le stagioni scegliendo il mese dell'anno voluto per la simulazione e permette di variare in maniera random le condizioni ambientali anche durante la simu-

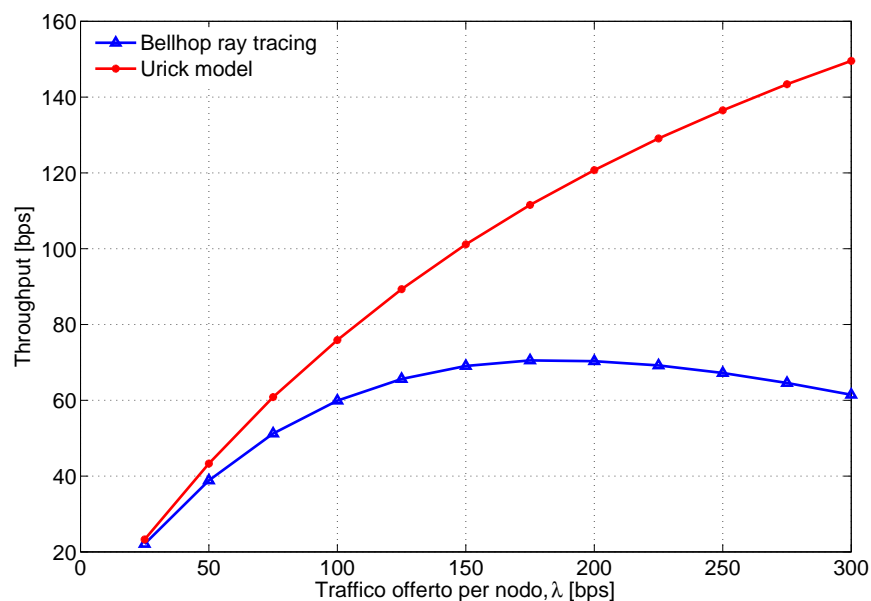


Figura B.1: Throughput in funzione del traffico offerto per il protocollo CSMA-ALOHA con i due modelli di canale disponibili.

lazione stessa. A titolo esemplificativo, si riporta in Figura (B.1) la differenza del throughput ottenuto con per il protocollo CSMA-ALOHA per i due modelli di canale disponibili in WOSS. È visibile come WOSS con Bellhop tenga conto di una interferenza maggiore che provoca un abbassamento del throughput a traffico crescente rispetto al modello semplificato di Urick in cui il throughput continua a crescere.

Bibliografia

- [1] C.R. Benson, M.J. Ryan, M.R. Frater “*Implications of simplifying SINR in Underwater Acoustic Networks*”,in *Proc. of MTS/IEEE Oceans*, Biloxi, MS, Oct. 2009
- [2] F. Guerra, P. Casari, M. Zorzi “*World Ocean Simulation System (WOSS): A Simulation Tool for Underwater Networks with Realistic Propagation Modeling*”, *WUWNet’09*, November, 3 2009
- [3] B. Peleato and M. Stojanovic “*A MAC Protocol for Ad-Hoc Underwater Acoustic Sensor Networks*”, *WUWNet’06*, pp. 113-115, September 25, 2006
- [4] M. Stojanovic “*On the Relationship Between Capacity and Distance in Underwater Acoustic Communications Channel*”,*WUWNet’06*, September 25, 2006
- [5] S. Azad, P. Casari, F. Guerra, M. Zorzi “*On ARQ Strategies over Random Access Protocols in Underwater Acoustic Networks*”,in *Proc. of IEEE/OES Ocean 11*, 2011
- [6] B. Peleato and M. Stojanovic “*Distance Aware Collision Avoidance Protocol for Ad-Hoc Underwater Acoustic Sensor Networks*”,*IEEE Comm. letters*, Vol. 11, NO. 12, December 2007
- [7] A. Syed, W. Ye, J. Heidemann “*T-Lohi: A New Class of MAC Protocols for Underwater Acoustic Sensor Networks*”,*IEEE in Proc. of INFOCOM*, pp. 789-797, 2008
- [8] *Ns2-miracle source code download page*. [Online]. Available: <http://telecom.dei.unipd.it/pages/read/58>

- [9] *World Ocean Atlas* [Online]. Available:
http://www.nodc.noaa.gov/OC5/WOA05/pr_woa05.html
- [10] *General bathymetric chart of the oceans*. [Online]. Available:
<http://www.gebco.net>
- [11] *National geophysical data center, seafloor superficial sediment description*. [Online]. Available: <http://www.ngdc.noaa.gov/mgg/geology/deck41.html>
- [12] *ns2 network simulator*, <http://www.isi.edu/nsnam/ns>
- [13] M. Porter *et al.*, “*Bellhop Code.*” [Online]. Available:
<http://oalib.hlsresearch.com/Rays/index.html>
- [14] M. Porter “*The BELLHOP Manual and User’s Guide: PRELIMINARY DRAFT*, Heat, Light and Sound Research, Inc. La Jolla, CA, USA, January 31, 2011
- [15] L. Freitag *et al.*, “*The WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms*”, in *Proc. of MTS/IEEE Oceans 2005*, 2005
- [16] L. F. M. Vieira *et al.*, “*Analysis of ALOHA protocols for underwater acoustic sensor networks*”, in *Proc. of ACM WUWNet*, Los Angeles, CA, Sep. 2006
- [17] A. Syed, Wei Ye, B. Krishnamachari, J. Heidemann, “*Understanding Spatio-temporal Uncertainty in Medium Access with ALOHA Protocols*”, in *Proc. of ACM MobiCom*, Montreal, Quebec, Canada, Sep 9-14 2007
- [18] F. Guerra, P. Casari e M. Zorzi, “*A performance comparison of MAC protocols for underwater networks using a realistic channel simulator*”, in *Proc. of MTS/IEEE Oceans*, Biloxi, MS, Oct. 2009
- [19] M.B. Porter, H.P.Bucker, “*Gaussian beam tracing for computing ocean acoustic fields*”, in *J. Acoust. Soc. America*, 1349-1359, 1987

Ringraziamenti

Sono quasi alla fine di questo lungo cammino e quasi non mi sembra vero. L'ho desiderato talmente tanto in questo ultimo periodo che quasi non me ne rendo conto. Al raggiungimento di un piccolo o grande traguardo, chiunque deve ringraziare chi l'ha sostenuto e aiutato e credo sia doveroso che anche io lo faccia. Innanzitutto un caloroso e sentito ringraziamento va ai miei genitori Mario e Aurora, per aver desiderato questo traguardo forse più di me e avermi spronato quasi fino alla noia. Un ricordo affettuoso va allo zio Adriano sempre generoso nei miei confronti e allo zio Roberto per avermi iniziato a suo modo al mondo dell'informatica e per avere pazienza quando gli prometto i lavori che non faccio mai.

Un ringraziamento va al mio relatore, prof. Michele Zorzi per avermi presentato questa possibilità di fare una tesi che per me è stata molto interessante. Un ringraziamento infinito va a Paolo Casari, mio supervisore e aiutante nella tesi. La sua esperienza, pazienza, fantasia, cordialità e il suo sorriso sono stati per me una grande scuola e sono fiero e sicuramente felice di aver lavorato con lui. Un ringraziamento grandissimo va a Federico Guerra, fonte inesauribile di consigli e di insegnamenti, direi che ne sa una più del diavolo. Un ringraziamento anche ai miei colleghi tesisti del gruppo underwater e agli altri componenti, in special modo a Beatrice e Riccardo. Un ringraziamento speciale anche a Saiful, che mi ha aiutato quotidianamente con grande pazienza, soprattutto nella prima parte, la più critica.

Un ricordo particolare ai miei compagni di avventura, veramente troppi da nominare singolarmente in questi sei anni di università.

Pur ricordando tutti, mi permetto di nominare Diego, mio compagno in mille progetti e mio compagno fin dal secondo giorno. Insieme abbiamo condiviso momenti felici e momenti impegnati, ma sempre abbiamo saputo prendere il tutto con il sorriso e abbiamo saputo divertirci nei momenti di svago. Grazie anche a Michael, Ale Zambon, Ale (anche se non lo perdonerò di aver lasciato gli studi troppo presto :)), Michele, Matteo e Massimo.

Un ringraziamento speciale ai membri del SIGNET per avermi accolto come uno di loro. In particolare grazie a Marco Mezzavilla, che mi ha introdotto nel gruppo, grazie anche per i numerosi passaggi a casa, parlando di fotografia, calcio e molto, molto altro. Grazie anche per avermi spronato a fare sempre meglio e per essersi sempre interessato al mio lavoro, aiutandomi sempre. Grazie anche ad Andrea Bardella, per avermi sempre spronato a far meglio e avermi aiutato ad inserirmi nel gruppo. Grazie anche a tutti quelli che non ho nominato ma con cui ho condiviso bei momenti.

Un grande ringraziamento va ai miei grandissimi amici, senza di loro probabilmente non starei scrivendo la mia tesi di laurea. Grazie per i momenti speciali passati insieme, grazie per tutte le pazzie fatte insieme e per le serate indimenticabili. Grazie Bepi, Ale, Davide con le rispettive morose. Grazie a Enrico, Agno, Nasa, Mauro, Andrea R. e Andrea Z. Spero di fare ancora tanta festa con voi e ricordate, il Fedo sta tornando, più forte di prima e con più voglia di far festa di prima!

Grazie al mio amico di sempre Luca, per aver condiviso con me ogni cosa, dalla più pazza alla più seria. So che sarà sempre vicino a me a gioire per i miei successi e ad aiutarmi quando ne avrò bisogno, di questo mi ritengo fortunato.

Dulcis in fundo, voglio ringraziare Chiara. Senza di lei probabilmente non sarei mai riuscito a fare tutto questo. Nei momenti più bui della laurea magistrale mi ha sempre aiutato in tutte le maniere possibili. È stata sempre comprensiva e prodiga di consigli preziosi. Grazie ai suoi gesti di generosità e di amore che mi hanno permesso di sorridere quando non ne avevo voglia e grazie per avermi sempre gratificato quando raggiungevo dei risultati e grazie anche per essersi sempre interessata ai miei problemi, restando ad ascoltarmi per ore intere.

Le sarò infinitamente grato e considero questo risultato in gran parte suo e sono fiero di avere una persona così speciale al mio fianco. GRAZIE A TUTTI!

Padova, 12 Dicembre 2011

Federico Favaro