

Práctica 2: Programación de Comunicaciones en MPI

Shamuel Manrique 802400 Aldrix Marfil 794976

08/01/2021

Resumen

En esta práctica se abordan algunos patrones de comunicación clásicos en paso de mensajes. La idea es caracterizar el comportamiento de dichos patrones y para ello se realizaron las siguientes dos actividades:

1. La primera tarea consistió en la implementación de estos patrones en lenguaje C utilizando MPI.
2. La segunda tarea consistió en la caracterización del comportamiento de estos patrones de comunicación.

Actividad 1: Implementación en C de comunicaciones en MPI

Para lograr los requerimientos de esta práctica se realizaron tres implementaciones:

- **Prueba de latencia:** Para esta sección se implementó el script **latency_test.c** el cual es un programa en MPI que mide la latencia de las comunicaciones por cada pareja asignadas a jugar Ping-pong (rebotar paquetes). La cantidad de envíos (rebotes) que se realizan se encuentra parametrizados.
- **Prueba de ancho de banda:** Para esta sección se implementó el script **bandwidth_test.c** el cual es un programa en MPI que mide el ancho de banda de un envío de N paquetes (a modo de ráfaga) de tamaño M entre dos procesos, un emisor y un receptor que emite un ACK de validación una vez recibido los N paquetes de la ráfaga.
- **Prueba de latencia de la operación de broadcast:** Para esta sección se implementó el script **broadcast_test.c** el cual es un programa en MPI que tiene dos funcionalidades:
 1. **Broadcast One-to-one:** Envío de mensajes usando Send/Receive en donde un proceso es el root y tiene que hacer el envío uno a uno al resto de procesos.
 2. **Broadcast Bcast:** Envío usando la función de Bcast de MPI el cual realiza el envío simultáneo a todos los procesos, solo se le indica el proceso raíz en la misma llamada.

Actividad 2: Caracterización de patrones

Caracterizar los patrones requiere obtener medidas de sus comportamientos. Para ello se calcularon las siguientes medidas: - Tiempos de ejecución - Velocidades de transferencia - Latencias de comunicación

A la hora de medir los tiempos se realizaron experimentos ejecutando varias repeticiones (ejecutar cada prueba varias veces) para minimizar la influencia de valores atípicos y de la falta de resolución del medidor de tiempo en las medidas.

Pruebas realizadas

Se realizaron diversas pruebas para experimentar distintos escenarios y observar cómo varían las distintas medidas de interés. Para ello, se utilizaron dos computadoras conectadas, cada una con 4 núcleos disponibles. Las pruebas realizadas involucran ejecuciones usando 4 procesos (mapeados en una misma máquina) y 8 procesos (mapeados en ambas máquinas). Respecto al tamaño de los paquetes que se envían los procesos tenemos 1 Byte, 1 Kilobyte, 1 Megabyte, 10 Megabyte.

Finalmente se calcularon las siguientes medidas para cada una de las implementaciones descritas en la actividad 1 y con los valores de parámetros mencionados previamente:

- **Prueba de latencia:**
 - **Tiempo de rebote:** Por cada pareja asignada se midió el tiempo de un rebote. Esto es, por ejemplo, para la pareja 0 - 1, se midió el tiempo que tarda el paquete en ser enviado desde 0 a 1 mas el tiempo del envío que tarda de 1 a 0.
 - **Tiempo total de rebote:** Como el número de rebotes puede variar, se midió el tiempo total que tardan todos los rebotes en ejecutarse por cada par de procesos.
- **Prueba de ancho de banda:**
 - **Tiempo de envío de paquetes:** Por cada par de parejas se midió el tiempo de recepción de una ráfaga de mensajes. Esto es, por ejemplo, para la pareja 0 - 1, se midió el tiempo que tarda el proceso 0 en enviar una cantidad N de paquetes y luego 1 envía un paquete confirmación al proceso 0 y se detiene la medición.
 - **Ancho de banda:** Teniendo la cantidad de paquetes enviados (N), el tamaño de su tipo de datos (T) y el tiempo en segundos (E) que tarda el envío de la ráfaga hasta su confirmación se puede calcular el ancho de banda como el máximo ancho de banda obtenido de sacar la siguiente cuenta para todos las rafagas de las parejas: $Bd = (N * T) / E$
 - **Velocidad de transferencia:** Esta velocidad de transferencia efectiva vendría dada por la fórmula anterior pero tomando el promedio de todas las cuentas $Bd = (N * T) / E$
- **Prueba de latencia de la operación de broadcast:** Se fijó el proceso 0 como root (es quien envía los paquetes a los otros nodos). Luego, se midió el tiempo que tardan los envíos del proceso root a los otros procesos para las siguientes dos implementaciones:

1. Broadcast One-to-one
2. Broadcast Bcast

Análisis de resultados de rendimiento

Prueba de latencia entre procesos en MPI

Se realizaron 10 ejecuciones del script de latencia. Y se emplearon variaciones de los parámetros Número de procesos (4 y 8), Tamaño del paquete enviado (1B, 10 KB, 1 MB, 10MB). Es importante recordar que acá se mide la cantidad de tiempo que tarda un paquete en rebotar entre un nodo A y un nodo B.

A continuación se mostrará el resumen estadístico de las latencias obtenidas para experimentos usando 4 y 8 procesos con envíos de paquetes de 1 Byte y 10 Megabytes.

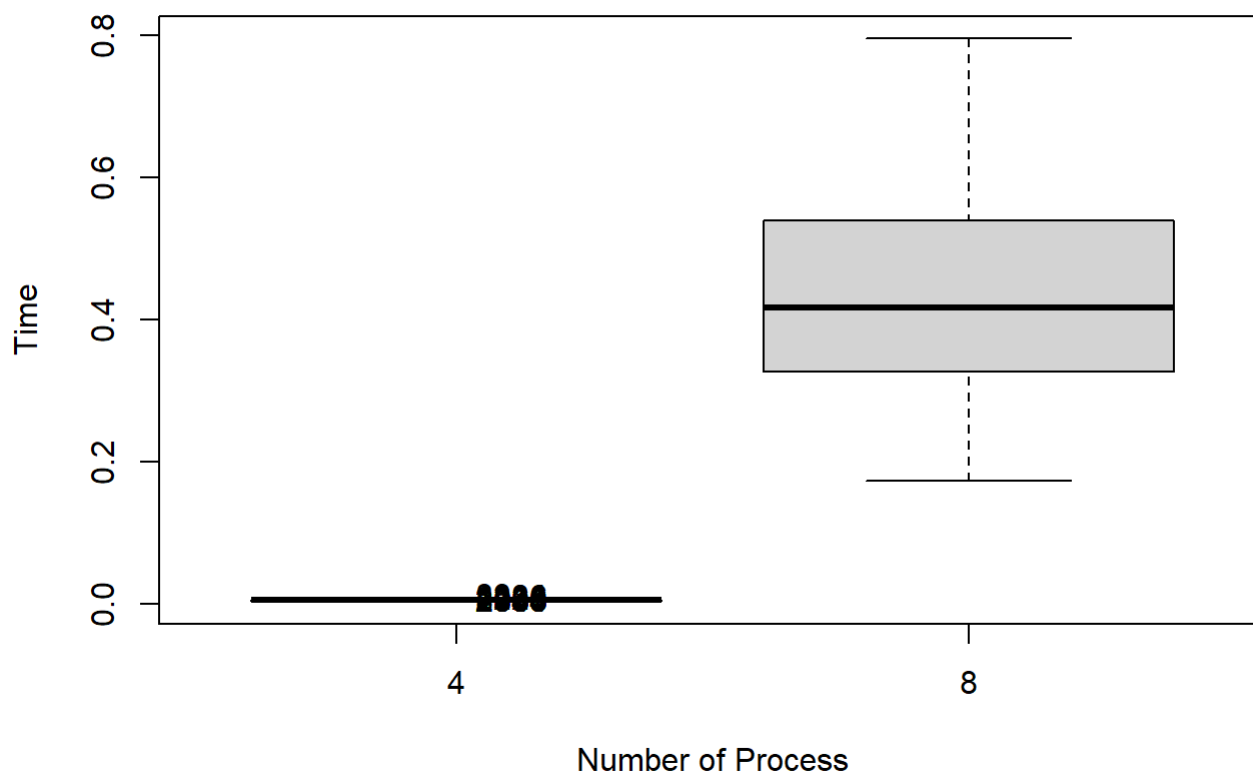
```
##          mean          sd          IQR skewness  kurtosis          0%
## 4 5.462716e-06 5.105899e-05 9.53700e-07 14.33066 214.93362 0.00000e+00
## 8 1.662505e-04 8.052691e-05 4.41075e-05 2.92069 18.74853 1.00136e-05
##          25%          50%          75%          100% COM_TIME:n
## 4 0.00000000000 0.0000009537 0.0000009537 0.0008749962          2040
## 8 0.0001299381 0.0001590252 0.0001740456 0.0009591579          4080
```

```
##          mean          sd          IQR skewness  kurtosis          0%
## 4 0.004393165 0.000569976 0.0006332994 -0.445495 7.7266041 0.002741098
## 8 0.436406213 0.134707772 0.2126019002 0.340637 -0.7814184 0.172389984
##          25%          50%          75%          100% COM_TIME:n
## 4 0.004174709 0.004316568 0.004808009 0.009623051          2040
## 8 0.326138854 0.416482449 0.538740754 0.794605970          4080
```

Se puede determinar por los resultados que el tiempo de envío y recepción del paquete depende del tamaño del paquete, a mayor tamaño mayor latencia. También se puede deducir que el número de procesos mapeados en distintas máquinas usadas para la ejecución influyen en el tiempo.

Para visualizar mejor la diferencia de latencia entre usar 4 y 8 procesos mapeados en distintas máquinas se muestra un diagrama de caja donde se aprecia el número de procesos en relación al tiempo de latencia. A partir de 4 procesos el mapeo de procesos se lleva a cabo entre dos máquinas. Por lo que el coste de comunicación es mucho mayor, Los procesos quedan distribuidos de forma tal que las parejas están separadas en máquinas distintas.

Gráfico de caja del tiempo promedio por número de proceso



Se observó que los datos están esparcidos y para la gráfica de boxplot se omiten valores atípicos para tener una visualización más legible y se procede a corroborar que existe una diferencia entre la ejecución del programa con todos los slots en una máquina o en distintas máquinas. Para poder comprobar si existe diferencia entre los grupos primero se realizó un test de normalidad donde se comprobó que ninguno de los datos sigue una distribución normal. También se validó que la varianza de los grupos son distintas. Realizada estas pruebas se empleó la prueba de wilcoxon donde se observa que el valor p se encuentra debajo de nuestro nivel de significancia ($\alpha=0,05$), con lo cual rechazamos la hipótesis nula (los grupos poseen la misma media) y concluimos que hay una diferencia estadísticamente significativa entre las dos medias.

Por lo tanto se puede concluir que la comunicación entre procesos usando MPI tiene un peso proporcional al número de máquinas distintas involucradas en el cálculo.

Métricas de latencia promediando los experimentos

A continuación se presenta una tabla resumen de los tiempos por tamaño del paquete y número de procesos

PACKET_SIZE <int>	NPROC <int>	BEST_TIME <dbl>	WORST_TIME <dbl>	AVERAGE <dbl>
1	4	0.0004	0.0015	0.0008
10000	4	0.0008	0.0024	0.0013
1000000	4	0.0191	0.0320	0.0254
10000000	4	0.4176	0.4760	0.4454
1	8	0.0153	0.0274	0.0188
10000	8	0.0513	0.0736	0.0604
1000000	8	3.5798	3.9293	3.7058
10000000	8	35.8210	48.3653	43.9104

8 rows

Observación: Estos resultados toman en cuenta el mejor y peor de los tiempos de entre todos los 10 experimentos realizados.

Tabla resumen del bandwidth y throughput

Ahora se muestra una tabla con el máximo y mínimo tiempo que toman los rebotes. También se muestra el máximo ancho de banda en cada caso (bandwidth) y la velocidad promedio de transferencia (throughput).

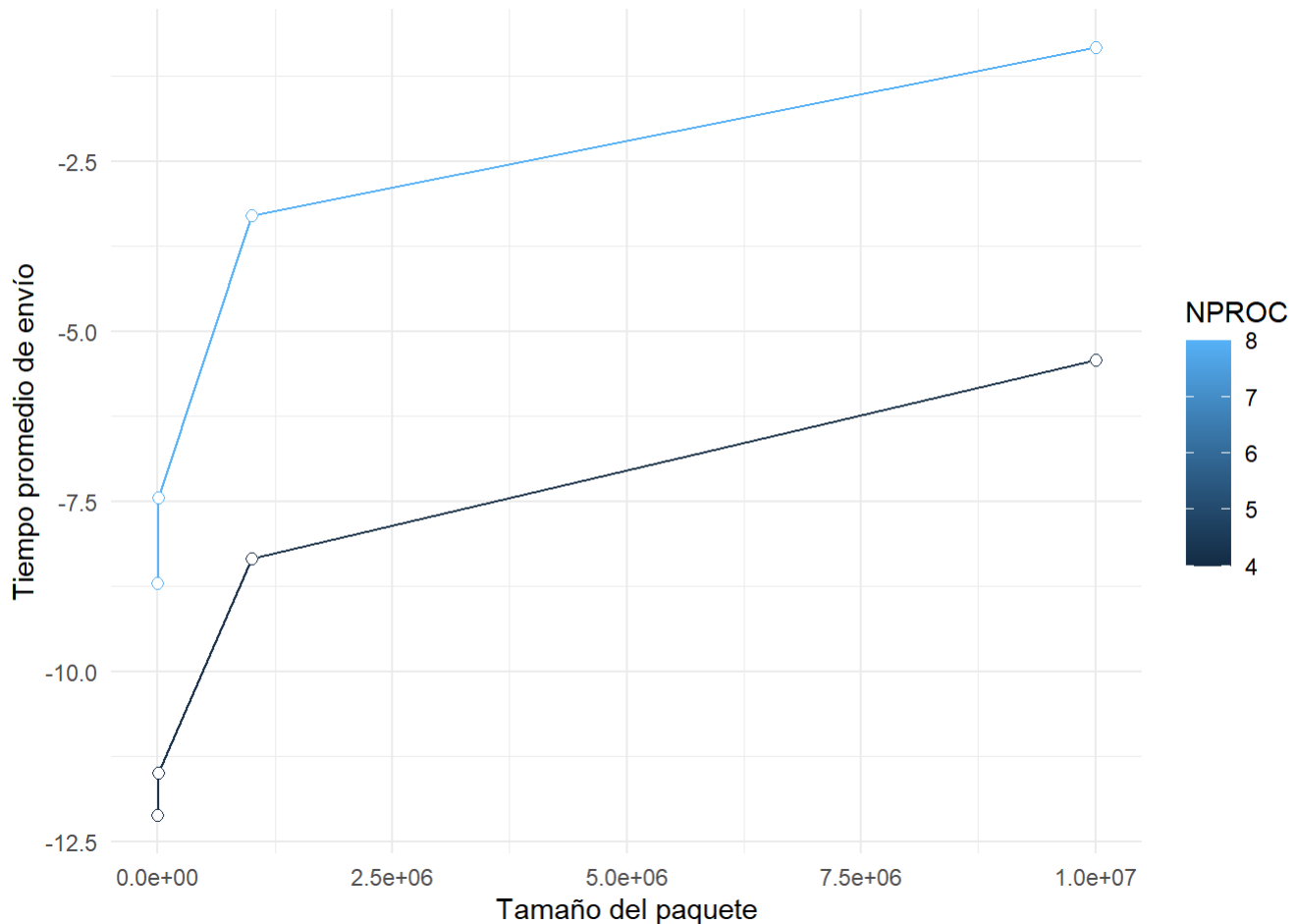
PACKET_SIZE <int>	NPROC <int>	BEST_TIME <dbl>	WORST_TIME <dbl>	AVERAGE <dbl>	BANDWIDTH <dbl>	THROUGHPUT <dbl>
1	4	0.0000000000	0.0008749962	5.462716e-06	NA	0.1746
10000	4	0.0000028610	0.0013740063	1.027957e-05	3333.360071	927.7380
1000000	4	0.0001749992	0.0013921261	2.387475e-04	5449.592435	3994.4899
10000000	4	0.0027410984	0.0096230507	4.393165e-03	3479.168484	2170.8140
1	8	0.0000100136	0.0009591579	1.662505e-04	0.095238	0.0057
10000	8	0.0000340939	0.0030679703	5.847967e-04	279.719925	16.3078

PACKET_SI...	NP...	BEST_TIME	WORST_TIME	AVERAGE	BANDWIDTH	THROUGH...
<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1000000	8	0.0181281567	0.0723240376	3.685145e-02	52.607352	25.8789
10000000	8	0.1723899841	0.7946059704	4.364062e-01	55.320750	21.8529

8 rows

Gráfica comparativa de tiempo de envío por tamaño del paquete distinguiendo por proceso

Se aplicó la función logaritmo a los ejes valores de la gráfica en ambos ejes para poder apreciar mejor la forma de la curva ya que los valores son muy pequeños.



Con las pruebas realizadas se puede apreciar que tanto el tamaño del paquete y el número de máquinas en la que se ejecutan los procesos influyen en los resultados obtenidos. El tiempo incrementa casi al doble al incluir nuevas máquinas. Se puede observar también que a partir de cierto umbral en el caso de 4 procesos `PACKET_SIZE=1000000` (1MB) y en 8 procesos `PACKET_SIZE=10000` (10KB) el bandwidth llega a su máxima capacidad y luego disminuye, esto se puede deber a que se requiere hacer más particiones del paquete para su envío.

Prueba de Bandwidth entre procesos en MPI

En esta oportunidad se realizaron 10 ejecuciones del script de bandwidth. Y se emplearon variaciones de los parámetros Número de procesos (4 y 8), Tamaño del paquete enviado (1B, 10 KB, 1 MB, 10MB). Es importante recordar que acá se mide la cantidad de tiempo que tarda una ráfaga de 100 paquetes en ser enviado desde un nodo A a un nodo B.

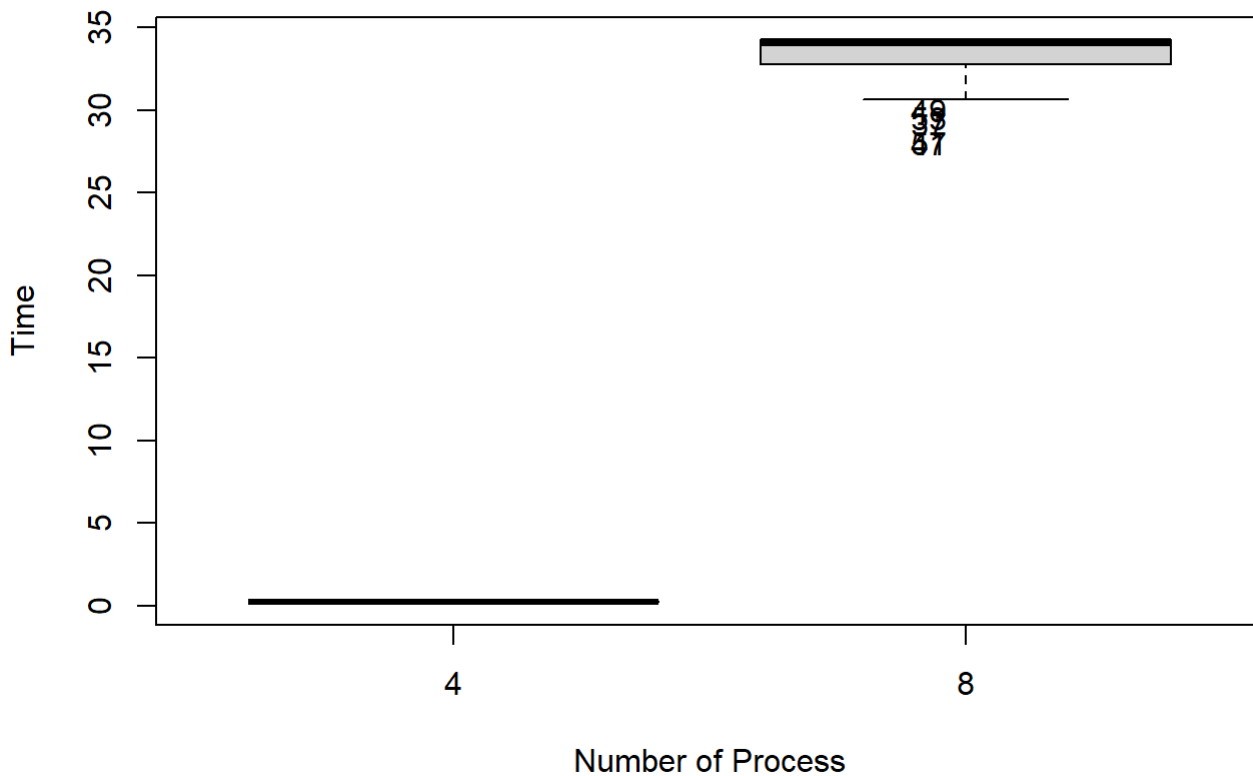
A continuación se mostrará el resumen estadístico de los tiempos obtenidos para experimentos usando 4 y 8 procesos con envíos de paquetes de 1 Byte y 10 Megabytes.

```
##          mean          sd          IQR    skewness    kurtosis
## 1      0.0001262188 0.0001261684 2.723938e-05 2.41550127 4.80872005
## 1000    0.0003133000 0.0001256864 1.595000e-04 1.26863195 0.28157334
## 10000   0.0003536224 0.0002218678 1.665950e-04 3.42439883 13.32134516
## 1000000 0.0093222022 0.0001620565 1.258850e-04 1.09306032 0.09920539
## 10000000 0.2164255500 0.0047404736 6.787419e-03 0.01822362 -0.89964954
##          0%          25%          50%          75%          100%
## 1      0.0000610352 0.0000696182 0.0000759363 9.685758e-05 0.000494957
## 1000    0.0002210000 0.0002265000 0.0002375000 3.860000e-04 0.000583000
## 10000   0.0002350807 0.0002427101 0.0002591610 4.093051e-04 0.001219988
## 1000000 0.00909999603 0.0092103481 0.0092815161 9.336233e-03 0.009645939
## 10000000 0.2086629868 0.2133557201 0.2156823874 2.201431e-01 0.224262953
##          COM_TIME:n
## 1          20
## 1000       20
## 10000      20
## 1000000    20
## 10000000   20
```

```
##          mean          sd          IQR    skewness    kurtosis
## 1      8.204758e-04 0.0001068089 0.0001673699 0.89442860 0.5478588
## 1000    4.047100e-03 0.0001303976 0.0002530000 0.02704959 -1.9551421
## 10000   3.323518e-02 0.0018399843 0.0033753514 -0.65225183 -0.7058398
## 1000000 3.385236e+00 0.0564467509 0.0498996973 -1.77609623 2.3730870
## 10000000 3.306088e+01 1.8729275891 1.4631003142 -1.70247771 1.7741342
##          0%          25%          50%          75%          100%
## 1      0.000658989 7.402896e-04 7.984638e-04 9.076595e-04 0.001146078
## 1000    0.003884000 3.924250e-03 4.035500e-03 4.177250e-03 0.004213000
## 10000   0.028655052 3.154707e-02 3.396142e-02 3.492242e-02 0.035173893
## 1000000 3.210985899 3.375120e+00 3.408554e+00 3.425020e+00 3.426824093
## 10000000 27.864249945 3.277951e+01 3.405979e+01 3.424261e+01 34.247819185
##          COM_TIME:n
## 1          40
## 1000       40
## 10000      40
## 1000000    40
## 10000000   40
```

Se puede determinar por los resultados que el tiempo de envío y recepción de la ráfaga de paquetes depende del tamaño del paquete, a mayor tamaño mayor tiempo. También el número de procesos mapeados en distintas máquinas usadas para la ejecución influyen en el tiempo.

Para visualizar mejor la diferencia de latencia entre usar 4 y 8 procesos mapeados en distintas máquinas se muestra un diagrama de caja donde se aprecia el número de procesos en relación al tiempo en recibir la ráfaga y confirmarla.



Se observa que existe una gran diferencia de tiempo respecto a la cantidad de procesos usados cuando están mapeados en distintas máquinas puesto que en la ráfaga de paquetes se envía bastantes datos comparado a la versión anterior de un paquete rebotando entre dos procesos.

Tabla comparativa de los resultados obtenidos para bandwidth

Ahora se muestra una tabla con el máximo y mínimo tiempo que toma enviar la ráfaga, recibir todos los paquetes y confirmar al emisor. También se muestra el máximo ancho de banda en cada caso (bandwidth) y la velocidad promedio de transferencia (throughput).

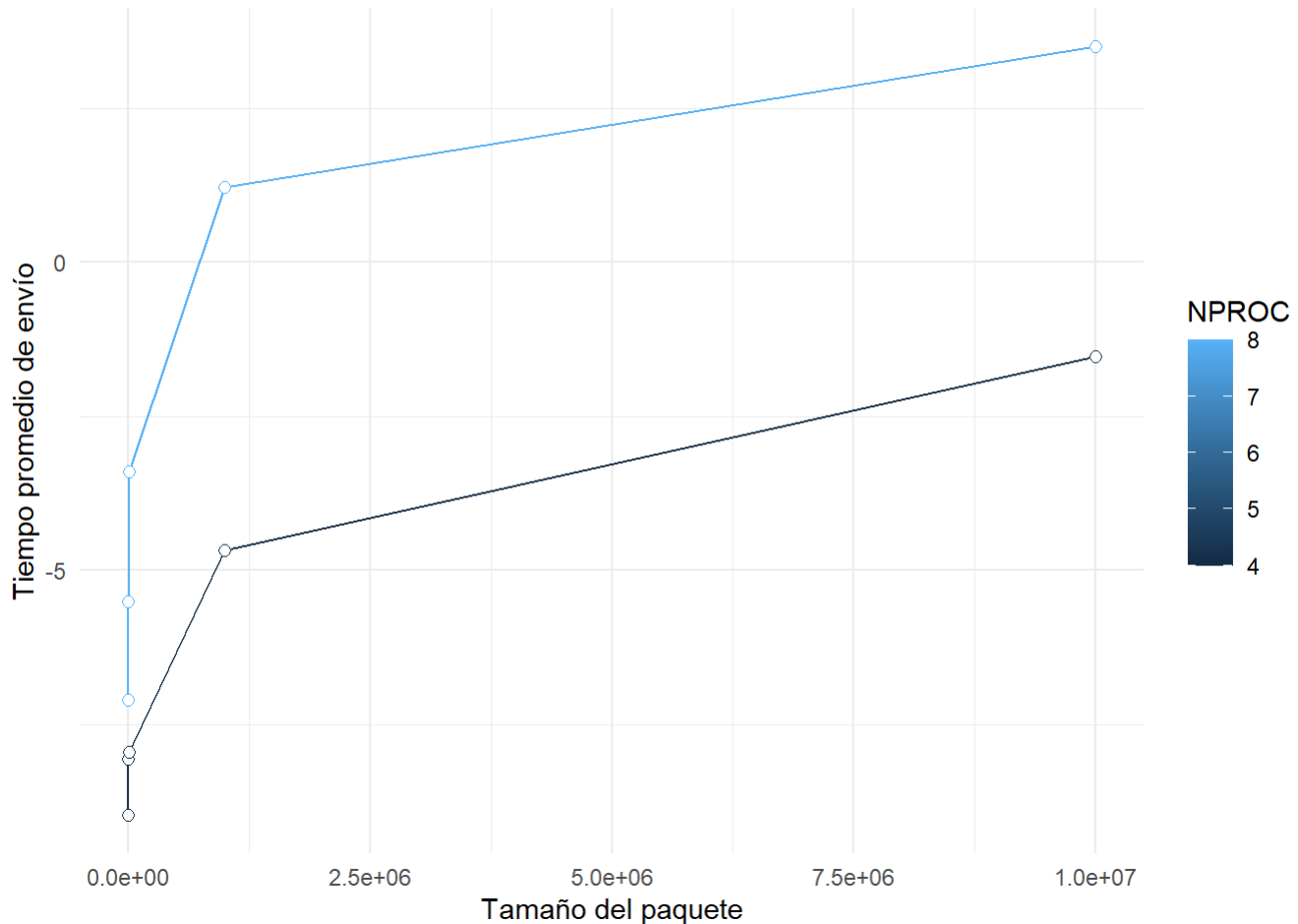
PACKET_SI...	NP...	BEST_TIME	WORST_TIME	AVERAGE	BANDWIDTH	THROUGH...
<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	4	6.103520e-05	0.000494957	1.262188e-04	0.015625	0.0076
1000	4	2.210000e-04	0.000583000	3.133000e-04	4.315268	3.0440
10000	4	2.350807e-04	0.001219988	3.536224e-04	40.567955	26.9687
1000000	4	9.099960e-03	0.009645939	9.322202e-03	104.799833	102.3014
10000000	4	2.086630e-01	0.224262953	2.164255e-01	45.704048	44.0648
1	8	6.589890e-04	0.001146078	8.204758e-04	0.001447	0.0012
1000	8	3.884000e-03	0.004213000	4.047100e-03	0.245539	0.2356
10000	8	2.865505e-02	0.035173893	3.323518e-02	0.332812	0.2869

PACKET_SI...	NP...	BEST_TIME	WORST_TIME	AVERAGE	BANDWIDTH	THROUGH...
<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1000000	8	3.210986e+00	3.426824093	3.385236e+00	0.297004	0.2817
10000000	8	2.786425e+01	34.247819185	3.306088e+01	0.342257	0.2885

1-10 of 10 rows

Gráfica comparativa de los tiempos por número de procesos

Se aplicó la función logaritmo a los ejes valores de la gráfica en ambos ejes para poder apreciar mejor la forma de la curva ya que los valores son muy pequeños.



Se puede observar que el comportamiento de latencia respecto a cuando se envía una gran rafaga de paquetes (100 en total) es similar al de los rebotes.

Prueba de Broadcast entre procesos en MPI

Envíos one to one

Resumen estadístico (4 procesos)

Se muestra una tabla resumen con los tiempos obtenidos realizando 10 pruebas con 4 procesos donde el nodo root asignado es el cero y era el encargado de enviar mensajes a todos los demás procesos.


```
##          mean          sd          IQR  skewness  kurtosis
## 1      2.439020e-05 9.384580e-06 0.0000009537 3.1287109 9.8396294
## 100    2.532006e-05 9.102895e-06 0.0000011325 3.0711094 9.5655376
## 1000   2.329351e-05 6.748315e-07 0.0000008941 -0.6115387 0.3067107
## 10000  3.089189e-04 9.257453e-05 0.0001412630 0.4526679 -1.3652795
## 100000 1.273942e-03 9.113932e-05 0.0000823140 1.9394127 4.5625014
## 1000000 6.196904e-03 1.385012e-04 0.0002333522 -0.1483797 -1.4934445
##          0%          25%          50%          75%          100%
## 1      0.0000209808 0.0000209808 2.098080e-05 0.0000219345 0.0000510216
## 100    0.0000209808 0.0000219345 2.253055e-05 0.0000230670 0.0000510216
## 1000   0.0000219345 0.0000229478 2.312660e-05 0.0000238419 0.0000240803
## 10000  0.0001850128 0.0002446174 2.639294e-04 0.0003858805 0.0004549026
## 100000 0.0011861324 0.0012206435 1.250505e-03 0.0013029575 0.0015020370
## 1000000 0.0060000420 0.0060938597 6.208062e-03 0.0063272119 0.0063638687
##          COM_TIME:n
## 1              10
## 100             10
## 1000            10
## 10000           10
## 100000          10
## 1000000         10
```

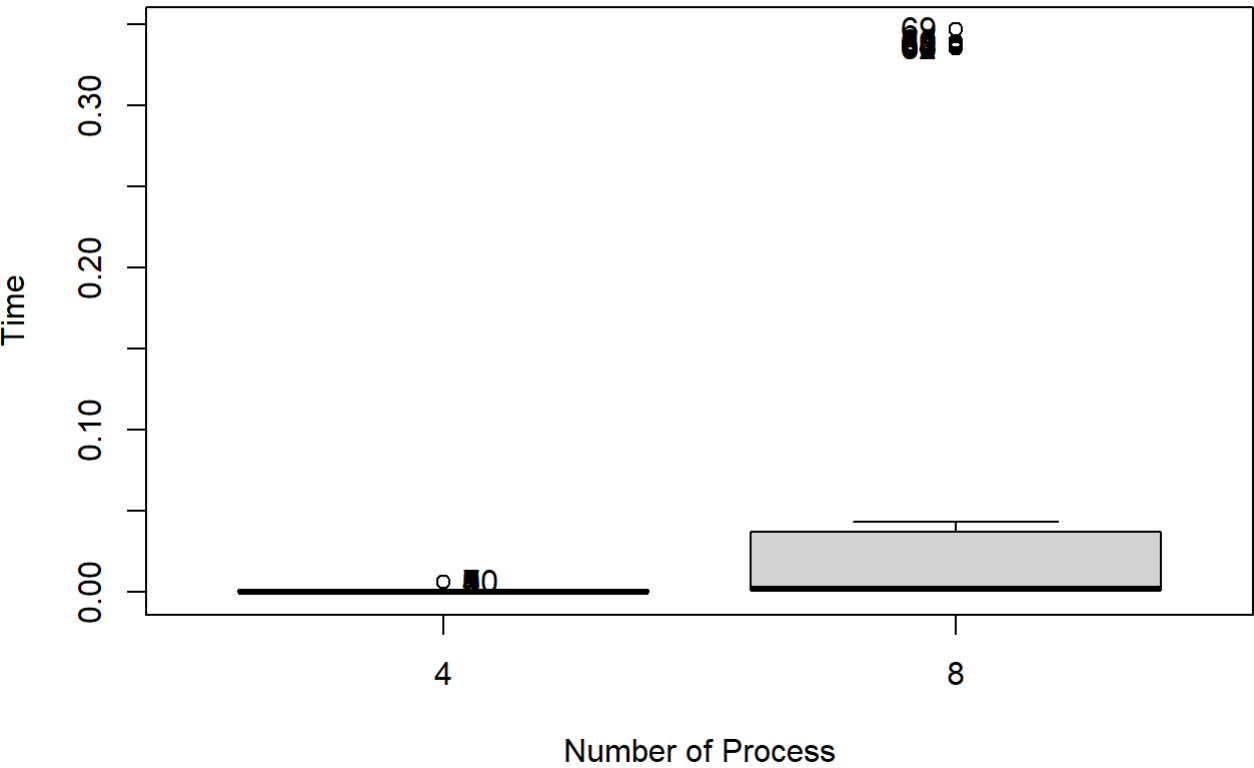
En resumen estadístico de la ejecución con 4 procesos se puede observar que el tiempo medio va incrementando con el tamaño del paquete, la desviación estándar se mantiene bastante alta. El valor del skewness nos indica que la distribución de la muestra es asimétrica positiva y tiene datos atípicos.

Resumen estadístico de haciendo envíos uno a uno (8 procesos)

```
##          mean          sd          IQR  skewness  kurtosis          0%
## 1      0.001566386 4.556345e-05 3.623958e-05 1.0278359 1.3194015 0.001509190
## 100    0.001641035 1.291872e-04 1.108050e-04 0.8304048 -0.1922280 0.001484871
## 1000   0.001662588 9.499338e-05 9.453295e-05 1.4270173 1.2606185 0.001570940
## 10000  0.002559113 1.324856e-04 1.931190e-04 0.8114782 -0.7080879 0.002408981
## 100000 0.037732458 2.193305e-03 1.289070e-03 2.4357563 6.7591816 0.035786152
## 1000000 0.337514138 3.502598e-03 3.194630e-03 2.0455026 4.9443749 0.334656954
##          25%          50%          75%          100% COM_TIME:n
## 1      0.001546681 0.001556039 0.001582921 0.001663923          10
## 100    0.001570582 0.001616955 0.001681387 0.001859903          10
## 1000   0.001605511 0.001622915 0.001700044 0.001868963          10
## 10000  0.002461076 0.002513528 0.002654195 0.002798080          10
## 100000 0.036597133 0.037183642 0.037886202 0.043562174          10
## 1000000 0.335034132 0.336675525 0.338228762 0.346388102          10
```

En resumen estadístico de la ejecución con 8 procesos vemos el mismo incremento de tiempos del caso anterior. Se puede resaltar que existe una menor desviación y no se posee tantos datos atípicos, sin embargo el tiempo de comunicación es mayor.

Gráfico de caja del tiempo promedio por número de procesos (PACKET_SIZE == 10000000)



Se puede apreciar el número de máquinas en los tiempos de comunicación al incrementar el número de procesos y máquinas involucradas..

Tabla resumen

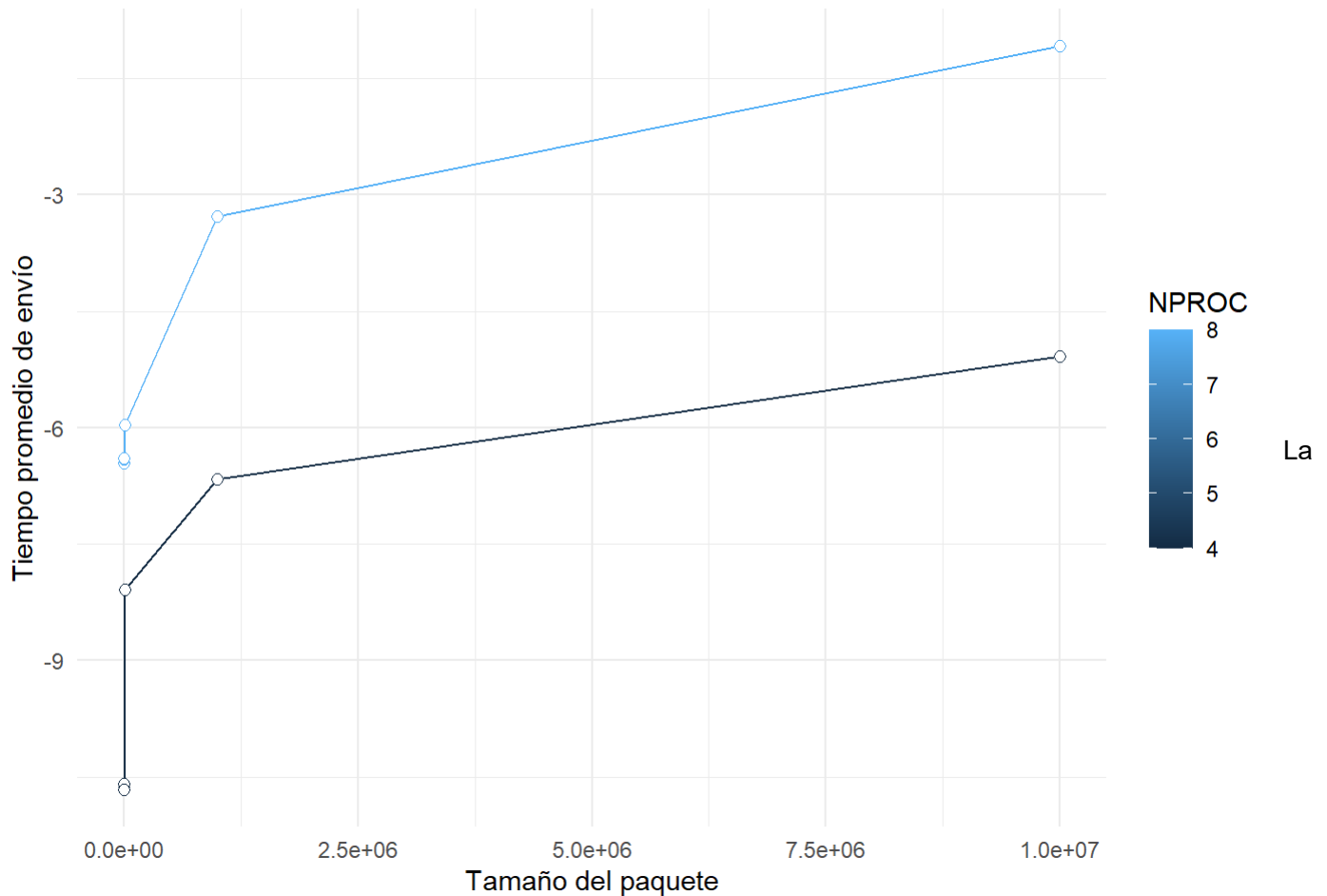
BCAST_T...	PACKET_S...	NP...	BEST_TIME	WORST_TIME	AVERAGE	BANDW...	THROU...
<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<c
0	1	4	0.0000209808	0.0000510216	2.439020e-05	0.0455	0.0
0	100	4	0.0000209808	0.0000510216	2.532006e-05	4.5455	3.7
0	1000	4	0.0000219345	0.0000240803	2.329351e-05	43.4783	40.9
0	10000	4	0.0001850128	0.0004549026	3.089190e-04	51.5464	30.8
0	1000000	4	0.0011861324	0.0015020370	1.273942e-03	804.0201	748.6
0	10000000	4	0.0060000420	0.0063638687	6.196904e-03	1589.4461	1538.9
0	1	8	0.0015091896	0.0016639233	1.566386e-03	0.0006	0.0
0	100	8	0.0014848709	0.0018599033	1.641035e-03	0.0642	0.0
0	1000	8	0.0015709400	0.0018689632	1.662588e-03	0.6071	0.5
0	10000	8	0.0024089813	0.0027980804	2.559113e-03	3.9588	3.7

1-10 of 12 rows

Previous12Next

Además de la influencia del número de máquinas en el tiempo se puede apreciar también que tanto el bandwidth como el throughput se ve afectado considerablemente por la comunicación remota.

Gráfica comparativa de tiempo de envío por tamaño del paquete distinguiendo por proceso



gráfica muestra la diferencia en tiempo de ejecutar el mismo programa con paquetes de distintos tamaños entre usando una cantidad variable de procesos.

Envío Bcast

Resumen estadístico de envíos usando la función Bcast de MPI (para 4 procesos)

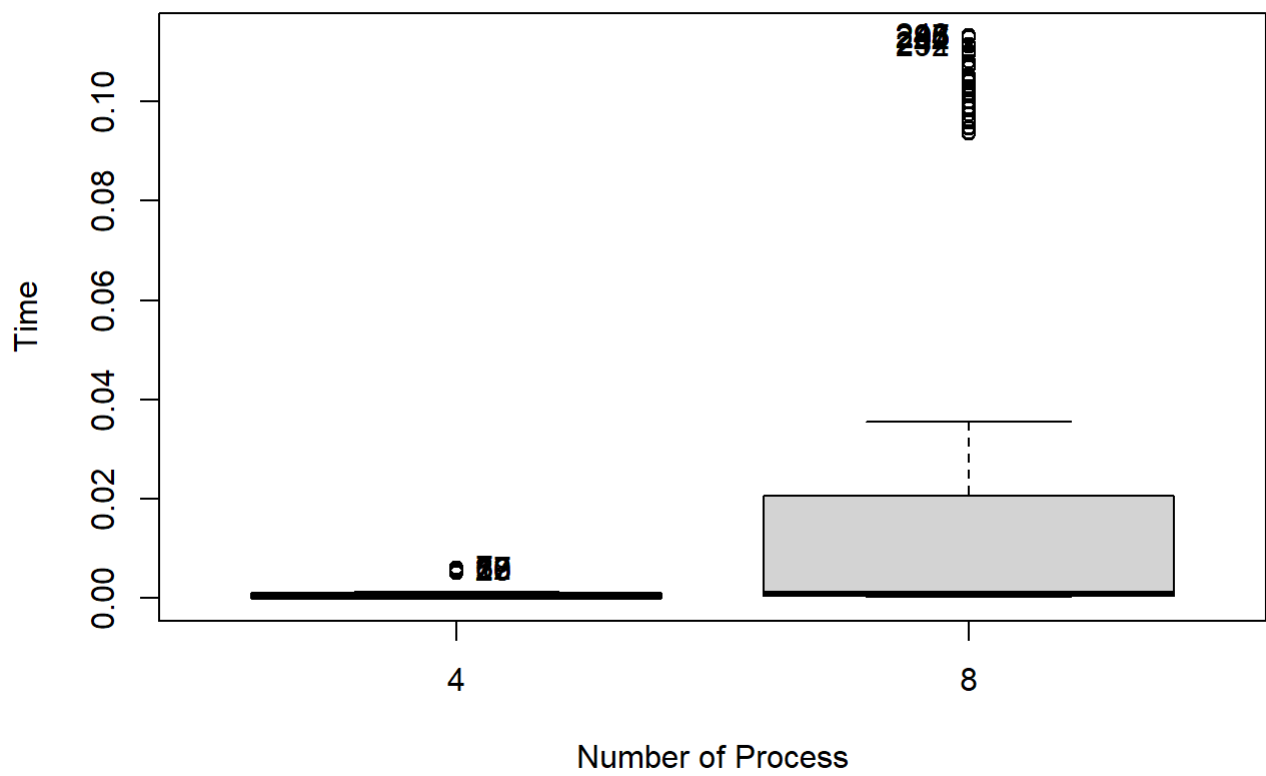
```
##          mean          sd          IQR    skewness    kurtosis
## 1      1.070857e-04 0.0001307979 0.0001181364 1.35894402 0.34979696
## 100    1.222193e-04 0.0001342209 0.0002081394 1.22097307 0.16309348
## 1000    9.015202e-05 0.0001144266 0.0000566244 2.11237737 3.89045757
## 10000   2.873719e-04 0.0001513291 0.0002270938 0.40012150 -1.26122530
## 100000  1.051825e-03 0.0001434425 0.0002763867 -0.08840169 -1.30408443
## 1000000 5.517107e-03 0.0002316418 0.0002768040 0.33314188 -0.02109007
##          0%          25%          50%          75%          100%
## 1      0.0000200272 2.503400e-05 3.337860e-05 1.431704e-04 0.0004761219
## 100    0.0000209808 2.765655e-05 4.601480e-05 2.357960e-04 0.0005028248
## 1000    0.0000219345 2.884868e-05 3.695485e-05 8.547307e-05 0.0004980564
## 10000   0.0000748634 1.885295e-04 2.239943e-04 4.156232e-04 0.0005509853
## 100000  0.0008411407 9.028912e-04 1.040578e-03 1.179278e-03 0.0012741089
## 1000000 0.0050849915 5.357623e-03 5.522132e-03 5.634427e-03 0.0060250759
##          COM_TIME:n
## 1              40
## 100             40
## 1000            40
## 10000           40
## 100000          40
## 1000000         40
```

Se observa cómo se reduce a más de la mitad el tiempo promedio de envío usando la librería Bcast, el tamaño del paquete influye también en el tiempo de los envíos. Se puede concluir que usando la librería Bcast el tiempo se reduce a la mitad en comparación a enviar uno a uno a cada uno de los procesos desde un nodo raíz.

Resumen estadístico de envíos usando usando la función Bcast de MPI (para 8 procesos)

```
##          mean          sd          IQR    skewness    kurtosis
## 1      0.0005734861 0.0001845879 1.084805e-04 1.4405536 2.3273049
## 100    0.0005542845 0.0001646875 9.447335e-05 0.8780581 2.7008637
## 1000    0.0005812615 0.0001234996 1.018047e-04 1.8697029 2.8201646
## 10000   0.0011878997 0.0003649841 4.581809e-04 -0.6187839 -0.6544398
## 100000  0.0222170234 0.0080216896 1.196927e-02 0.1703603 -1.2561064
## 1000000 0.1042953998 0.0054557969 8.337498e-03 0.1023311 -0.8800620
##          0%          25%          50%          75%          100%
## 1      0.0002388954 0.0004889965 0.0005315542 0.0005974770 0.001055956
## 100    0.0001759529 0.0004937649 0.0005400180 0.0005882382 0.001045227
## 1000    0.0004441738 0.0004947186 0.0005619526 0.0005965232 0.000921011
## 10000   0.0003039837 0.0009967685 0.0012804269 0.0014549493 0.001741886
## 100000  0.0102200508 0.0168507695 0.0206329823 0.0288200379 0.035460949
## 1000000 0.0936560631 0.1013380289 0.1030058861 0.1096755266 0.113250017
##          COM_TIME:n
## 1              80
## 100             80
## 1000            80
## 10000           80
## 100000          80
## 1000000         80
```

Gráfico de caja del tiempo promedio de envío por número de procesos usando Bcast (tamaño de paquete 10MB)



Según las mediciones obtenidas no hay una diferencia significativa en usar la función Bcast MPI o el envío uno a uno, al menos para menos de 8 procesos. Se aprecia en el siguiente gráfico de caja como las medias de las mediciones para cada grupo (4 y 8 procesos) son similares.

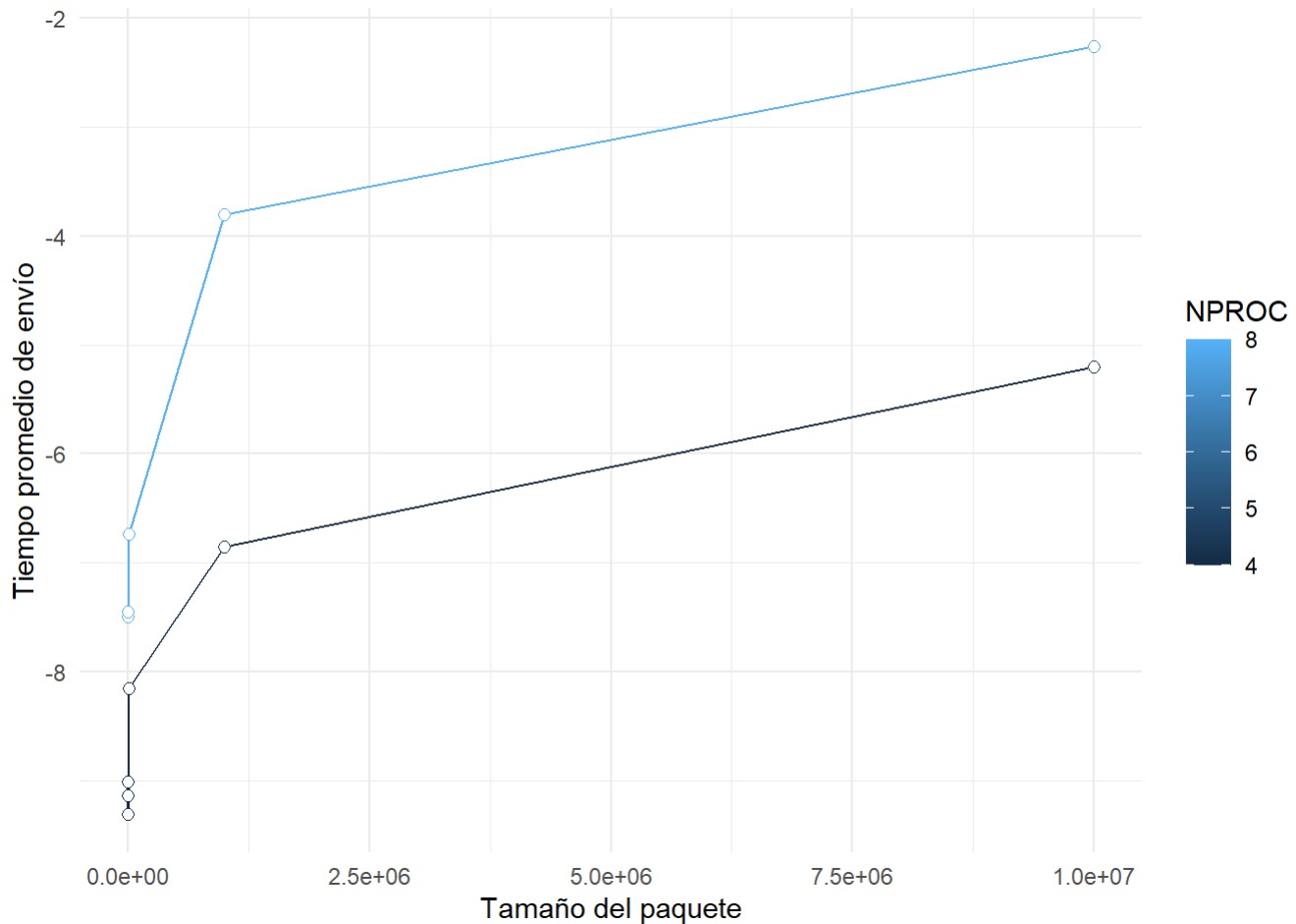
Tabla resumen (Tiempos y Datos transmitidos)

BCAST_T...	PACKET_S...	NP...	BEST_TIME	WORST_TIME	AVERAGE	BANDW...	THROU...
<int>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<c
1	1	4	0.0000200272	0.0004761219	1.070857e-04	0.0476	0.0
1	100	4	0.0000209808	0.0005028248	1.222193e-04	4.5455	0.7
1	1000	4	0.0000219345	0.0004980564	9.015202e-05	43.4783	10.5
1	10000	4	0.0000748634	0.0005509853	2.873719e-04	127.3886	33.1
1	1000000	4	0.0008411407	0.0012741089	1.051825e-03	1133.7869	906.6
1	10000000	4	0.0050849915	0.0060250759	5.517107e-03	1875.4689	1728.5
1	1	8	0.0002388954	0.0010559559	5.734861e-04	0.0040	0.0
1	100	8	0.0001759529	0.0010452271	5.542845e-04	0.5420	0.1
1	1000	8	0.0004441738	0.0009210110	5.812615e-04	2.1471	1.6
1	10000	8	0.0003039837	0.0017418861	1.187900e-03	31.3725	8.0
1-10 of 12 rows							
Previous 1 2 Next							

Se observan diferencias en tiempo en el orden de las diezmilésimas (cuarto decimal) para los mejores tiempos en los diferentes tamaños de paquetes. Luego, el máximo ancho de banda obtenido fue de 1875,47 MB por segundo con una transferencia promedio de datos de 1728.57.

Es importante resaltar que el ancho de banda se ve influenciado por la cantidad de procesos usados, puesto que mientras más procesos se usen va disminuyendo el ancho de banda disponible.

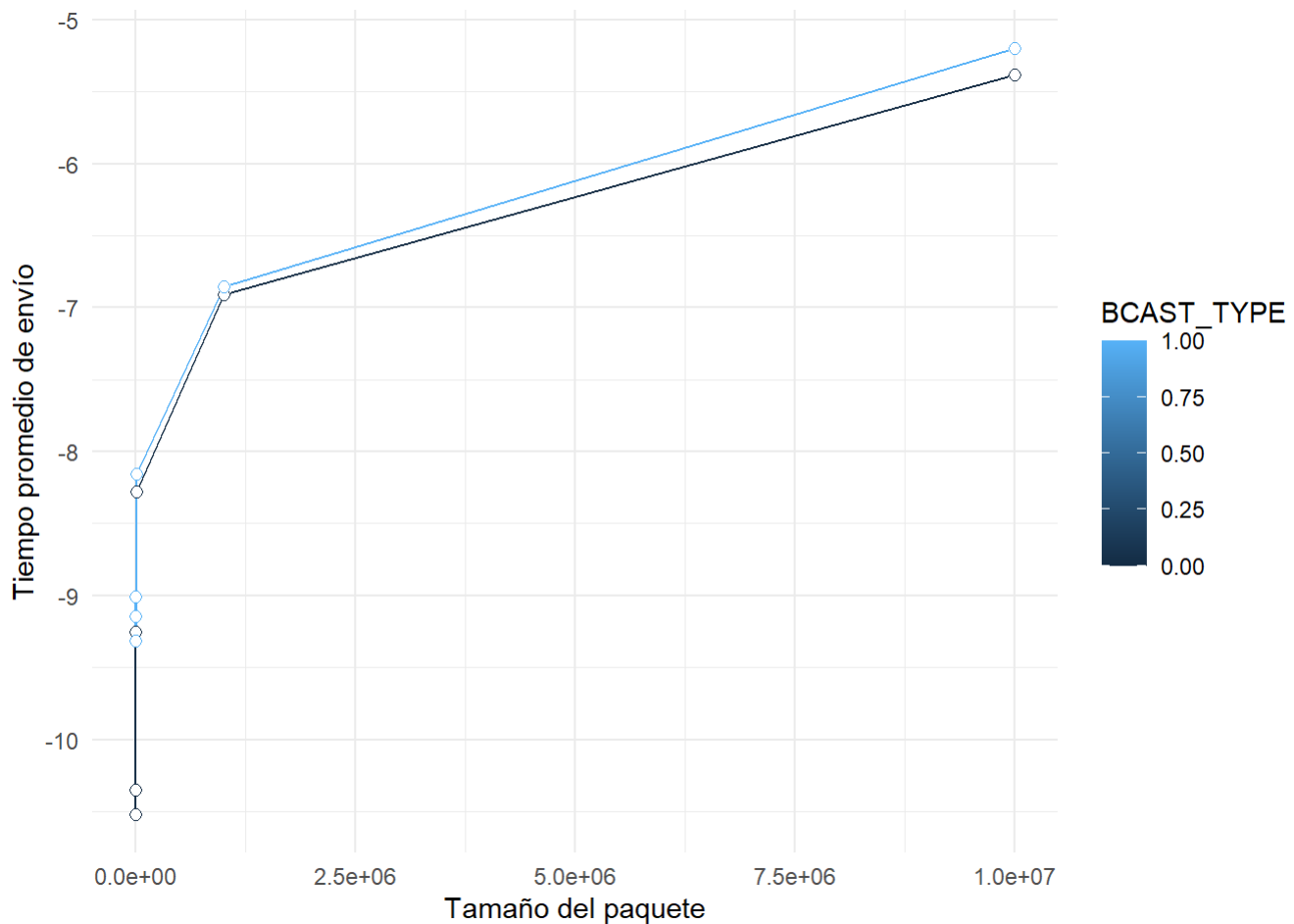
Gráfica comparativa de tiempo de envío por tamaño del paquete distinguiendo por proceso



Comparativa

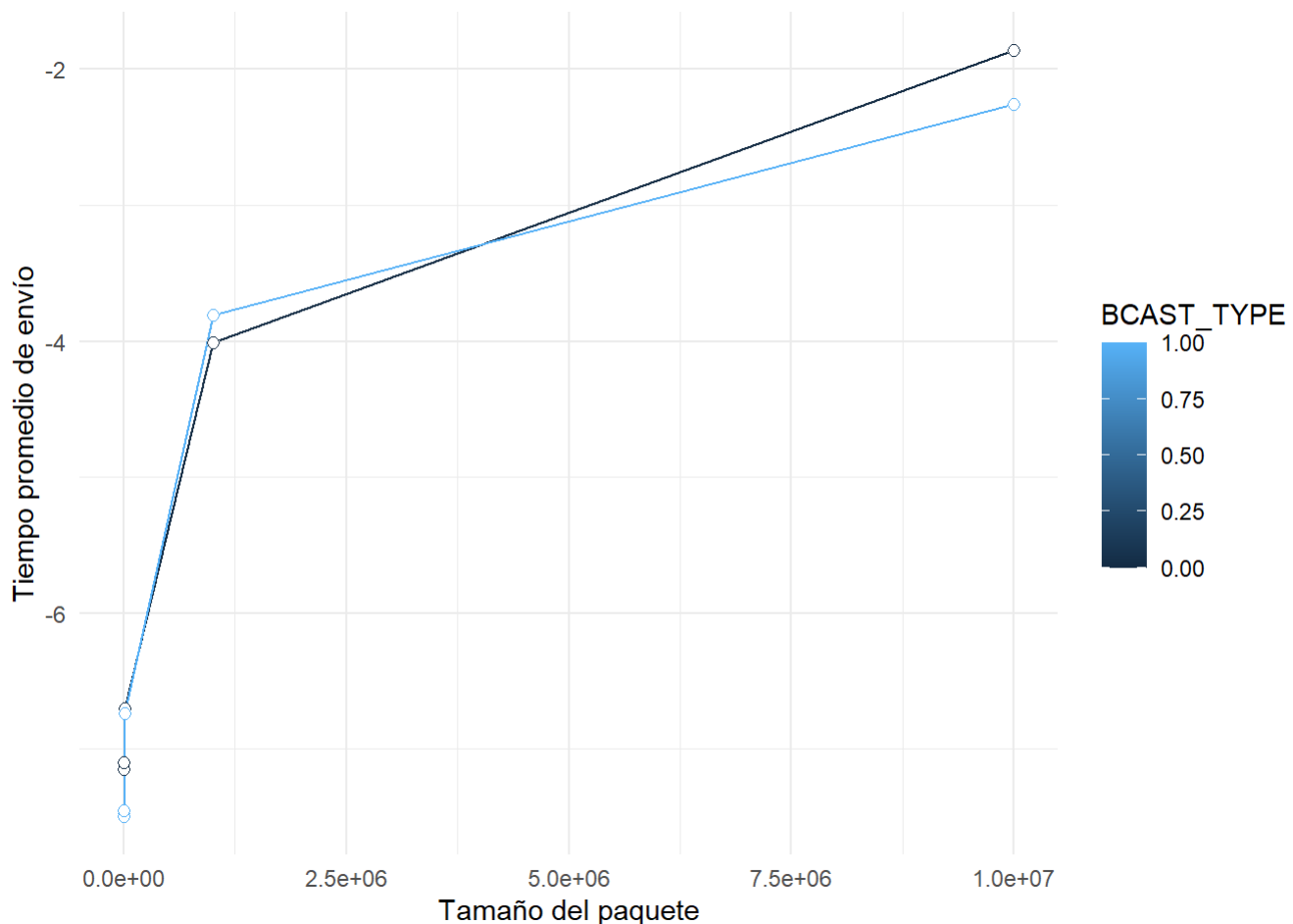
Por los resultados obtenidos con los tiempos de (Bcast 1, one-to-one 0) y graficando para una mejor visualización de las diferencias.

Comparativa bcast vs one to one (4 procesos)



Respecto a los tiempos de no se aprecia una mayor diferencia con cuatro procesos entre ambas métodos. Por los resultados obtenidos en particular no correspondían con lo esperado dado que se tendería creer que la implementación de MPI de Bcast debe ser más eficiente en estos casos, pero los resultados obtenidos nos dicen todo lo contrario.

Comparativa bcast vs one to one (8 procesos)



El desempeño de la versión MPI de broadcast es bastante similar al del envío uno a uno empleando varios procesos siempre y cuando el tamaño de los paquetes sea pequeño. Por ejemplo, en la gráfica se observa que para un paquete de 10 MB si hay una brecha en la duración del envío a todos vs el envío uno a uno a favor del Bcast de MPI. Posiblemente, esta función de MPI está pensada para sacar mayor partido al ejecutar el programa usando un mayor número de procesos.

Apendice:

Archivos de ejecución

Para la ejecución de forma automatizada y parametrizada de cada uno de los script de C se crearon bash scripts para automatizar los experimentos. Un ejemplo de los siguientes valores especificados para la ejecución son:

```
PROGRAM="name.out"      -> Nombre del archivo compilado con mpicc.
CSV_NAME="name_prueba"  -> Nombre del archivo csv con los resultados.
HOSTFILE="host_name.txt" -> Nombre del Host File a usar
PACKET_SIZES=(500)      -> Arreglo con los distintos tamaños del Paquete
NUMBER_PACKETS=(1000)   -> Arreglo con los distintos número de prueba/ejecuciones.
NUMBER_PROCCES=(4 8 16) -> Arreglo con los distintos número de procesos.
```

Archivo resultante de las ejecuciones

Para facilitar el análisis de los resultados obtenidos de tiempo de ejecución, velocidad de transferencia y latencia de comunicación por cada uno de los scripts se guarda un archivo de salida .csv con la siguiente cabecera:

```
BCAST_TYPE    -> Tipo de broadcast en caso de aplicar (solo aplica a los pruebas de broadcast).
PACKET_SIZE   -> Tamaño en Bytes del paquete.
NPROC         -> Cantidad de procesos.
N_PACKETS     -> Cantidad de paquetes a enviar.
N_BOUNCES     -> Número de rebotes del paquete de un proceso a otro (solo aplica para la prueba d
e latencia).
NODE          -> Máquina donde se ejecuta el proceso.
PROCESS       -> Identificador(Rank) del proceso.
SRC           -> Proceso que envía el/los mensaje/s
DST           -> Proceso que recibe el/los mensaje/s
TAG           -> Identificador único de parejas conectadas.
COM_TIME      -> Uso MPI_Wtime para medir el tiempo de envío y confirmación de recepción.
RUNING_TIME   -> Tiempo de ejecución del proceso.
```

Por mantener consistencia en todos los .csv dependiendo del script que se ejecute puede mostrar alguna columna vacía dado que ese dato no es requerido para ese script en particular. Por otra parte el PACKET_SIZE como se está usando tipo entero (4 Bytes) se multiplica el tamaño introducido por cuatro bytes para obtener la cantidad total en bytes, las medidas de tiempo usadas COM_TIME y RUNING_TIME están expresadas en segundos.