



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش پروژه

درس شبکه های کامپیوتری

دکتر حجازی

سید حسن سعادت میرقدیم

۹۶۳۱۰۴۰

• تعاریف

انواع گره ها: گره ها با توجه به ماهیت کارکردی خود به دو دسته ی گره های درونی یا ساده و گره های دروازه ای یا Gateway Node تقسیم می شوند. تفاوت گره های درونی با گره های دروازه ای در این است که این گره ها تنها به گره های درون خوشه دسترسی مستقیم دارند و به بیرون خوشه دسترسی مستقیم ندارند چون آدرس آی پی عمومی یا Public در اختیار ندارند اما گره های دروازه از یک طرف آدرس آی پی خصوصی و از طرف دیگر آدرس عمومی در اختیار دارند و میتوانند با سایر گره های دروازه در خوشه های مختلف ارتباط برقرار کنند، همچنین وظیفه تبادل و جابجایی داده بین گره های درون خوشه با گره های خارج از آن را بر عهده دارند. گره های درونی برای گره های سایر خوشه ها (چه گره های درونی و چه گره های دروازه) قابل دیدن نیستند و تمامی ارتباطات با بیرون از خوشه از گره دروازه می گذرد.

*تنظیم نوع گره از طریق سویچ g- یا gateway- که یک پرچم(flag) است امکان پذیر است.

• فرضیات

- ۱- هر خوشه فقط یک گره دروازه دارد و تعریف بیش از آن نتایج غیر قابل پیش بینی به وجود می آورد.
- ۲- در اینجا فرض را بر این گرفتیم که لیست هر گره اطلاعات آن گره را شامل نمی شود اما در فرایند ارسال آن را نیز به سایر گره ها در همان قالب ارسال لیست می فرستد.
- ۳- نام گره ها در تمام خوشه ها منحصر به فرد است و در دو خوشه متمایز هیچ دو گره همنامی یافت نمی شود. مثلاً در گره ۱ گره ها اسامی N_1 تا N_k را در اختیار دارند و در گره دوم اسامی از N_{k+1} تا N_p می باشد و الی آخر.

• پروتکل تدوین شده

قالب کلی پیام ها:

CMD	:	SRC	:	DST	:	*	:	*
-----	---	-----	---	-----	---	---	---	---

CMD → Command

: → Delimiter

SRC → Source Node

DST → Destination Node

* → Variable command by command

انواع پیام های ارسالی بر حسب دستور یا کامند:

۱- پیام اکتشاف یا Discovery:

با توجه به پیاده سازی قسمت امتیازی این پروژه، این نوع پیام بسته به نوع گره به سه نوع پیام اکتشاف مرزی (Border Discovery)، دروازه ای (Gateway Discovery) و ساده (Discovery) تقسیم می شود. پیام اکتشاف دروازه ای پیام هایی است که از طرف گره دروازه به سمت گره های درون خوشه فرستاده می شود و عکس آن پیام اکتشاف مرزی است که به سمت گره های دروازه ی سایر خوشه ها ارسال می گردد. پیام اکتشاف ساده نیز از طرف گره های درونی به سمت گره دروازه یا سایر گره های درونی ارسال می گردد.

DISCOVERY	:	SRC	:	DST	:	<list>
-----------	---	-----	---	-----	---	--------

List: رشته ای است حاوی لیست گره های ارسالی که توسط متد discovery_message_serializer تولید می شود.

نمونه هایی از این پیام:

- DISCOVERY:N2:N1: N3,127.0.0.1,8000|N2,127.0.0.1,8000|N3,127.0.0.1,8002
- BORDER_DISCOVERY:N1:N4: N1,192.168.0.1,8000| N2,192.168.0.1,8000| N3,192.168.0.1,8000.....
- GATEWAY_DISCOVERY:N1:N2: N4,127.0.0.1,8000|N2,127.0.0.1,8001|N3,127.0.0.1,8002

همانطور که در مثال های فوق مشاهده می شود نوع آدرس های ارسالی در این پیام ها با یکدیگر متفاوت است که در فرایند اکتشاف مفصلاً تشریح می شود.

۲- پیام درخواست فایل:

GET	:	SRC	:	DST	:	<filename>
-----	---	-----	---	-----	---	------------

این پیام از سوی گره درخواست دهنده به گره های لیست ارسال می گردد و آن ها با بررسی فایل های موجود به آن پاسخ مناسب می دهند. (ر.ک. دو پیام بعدی)

۳- پیام تایید یا ACK:

گره درخواست شونده پس از اطلاع از موجود بودن فایل در پوشه خود اقدام به ارسال این پیام به گره درخواست دهنده می نماید. قالب این پیام در زیر آمده است:

<tcp_port>	:	<filename>	:	DST	:	SRC	:	ACK
------------	---	------------	---	-----	---	-----	---	-----

گره پاسخ دهنده نام فایل درخواستی و همچنین شماره پورت TCP خود را برای درخواست دهنده ارسال می کند. علت ارسال نام فایل اینست که اگر پاسخ به موقع به دست گره درخواست دهنده نرسید (به علت تاخیر احتمالی در شبکه) آن گره آن را از سایر درخواست فایل ها تشخیص دهد.

۴- پیام رد یا NACK:

گره درخواست شونده پس از اطلاع از موجود نبودن فایل در پوشه خود اقدام به ارسال این پیام به گره درخواست دهنده می نماید. قالب این پیام در زیر آمده است:

<filename>	:	DST	:	SRC	:	NACK
------------	---	-----	---	-----	---	------

گره پاسخ دهنده نام فایل درخواستی را برای درخواست دهنده ارسال می کند. علت ارسال نام فایل اینست که اگر پاسخ به موقع به دست گره درخواست دهنده نرسید (به علت تاخیر احتمالی در شبکه) آن گره آن را از سایر درخواست فایل ها تشخیص دهد. مسلماً ارسال شماره پورت TCP در اینجا لزومی ندارد بنابراین در ساختار پیام گنجانده نشده است.

• نخ ها یا Threads

این برنامه در چهار نخ اصلی اجرا می شود (در صورت انتقال فایل دو یا بیشتر نخ دیگر نیز اضافه می شود)

۱- نخ سرویس دهنده ی سرور TCP

این نخ دائماً در حال گوش دادن به درخواست های دریافت فایل است و به محض دریافت تقاضا برای سرویس دادن به آن نخ مجزایی ایجاد کرده و به ادامه گوش دادن می پردازد.

۲- نخ سرویس دهنده خط فرمان

دستورات کاربر را پردازش و اجرا می کند. ۳ دستور داریم الف) دستور list که لیست گره ها را نشان می دهد ب) دستور get که پارامتر ورودی آن نام فایل است و به گره های لیست تقاضای دریافت آن فایل را می فرستد.

✓ get <filename>

پ) دستور left که نخ ها را پایان می دهد و برنامه را به اتمام می رساند.

۳- نخ سرویس ارسال اکتشاف

در فاصله ی زمانی مشخص پیام اکتشاف به گره ها ارسال می کند.

۴- نخ سرویس دهنده ی سرور UDP

پیام های UDP را پاسخ می دهد و یا forward می کند.

• فرایند اکتشاف یا Discovery

در فرایند اکتشاف نیاز است که هر گره به طور مجزا لیستی از گره ها در اختیار داشته باشد و به صورت خودکار و دوره ای اقدام به ارسال آن به سایر گره ها کند و از طرفی لیست هایی را نیز از سایر گره ها دریافت نموده و با لیست خود ادغام کند.

فرضیات:

قالب نگه داری لیست در فایل برای هر گره به صورت زیر می باشد:

نام	آدرس	پورت UDP
-----	------	----------

به عنوان مثال:

N1	192.168.0.1	8000
----	-------------	------

مشابه همین قالب برای ذخیره لیست گره ها به صورت دیکشنری (در متغیر cluster_list) با کلید نام و مقدار (پورت, آدرس) در کد استفاده می گردد.

فرایند اکتشاف در دو نخ (Thread) مجزا از یکدیگر دنبال می شود. یکی ارسال کننده پیام اکتشاف یا discovery_sender و دیگری دریافت کننده پیام اکتشاف به اسم udp_server_handler که علاوه بر دریافت اکتشاف وظیفه دریافت سایر پیام های udp را نیز دارد و به همین سبب نامش متفاوت است.

نخ اول در یک حلقه بینهایت که با پرچم DSC_SEND کنترل می شود در بازه های زمانی به اندازه disc_inter - به معنای Discovery interval که متغیری برای تنظیم فاصله ارسال پیام های اکتشاف است -

لیست گره هایی (به انضمام آدرس خود گره) که در اختیار دارد را به وسیله ی متد discovery_message_serializer به صورت رشته ای با فرمت زیر در آورده و به گره های لیست ارسال می کند:

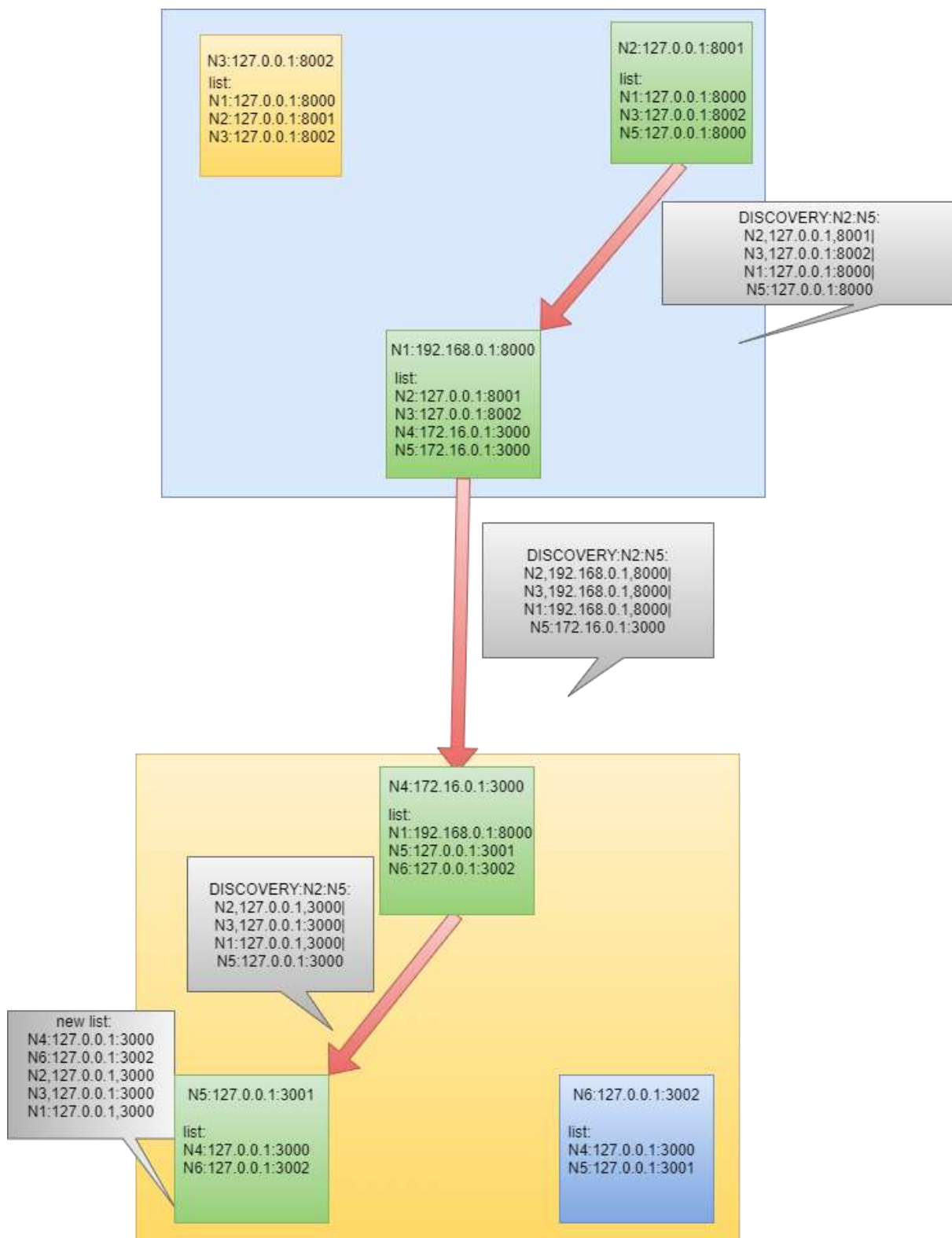
N4,192.168.0.1,8000|N2,127.0.0.1,8000|N3,127.0.0.1,8002

*نقطه ها به معنی ادامه لیست هستند و جزو پروتکل نیستند.

این رشته در قالب پیام نوع ۱ قرار گرفته و ارسال می گردد.

در اینجا اگر گره دروازه بخواهد لیست خود را برای گره های درونی خوشه خویش ارسال کند، لازم است تا آدرس گره های خوشه های دیگر را به آدرس خصوصی خود تغییر دهد تا گره های درونی درخواست های خود به گره های بیرونی را از طریق این گره ارسال کنند. تعیین نوع پیام اکتشاف - ر.ک. پیام ۱ - در اینجا صورت می گیرد.

نخ دوم نیز وظیفه دریافت پیام های اکتشافی و ادغام آنها با لیست گره ها را دارد. از طرفی اگر گره از نوع دروازه باشد آنگاه وظیفه انتقال (forward) پیام ها (چه اکتشافی، چه درخواست فایل و چه انتقال آن) از گره های درون به بیرون و بالعکس را دارد. مکانیزمی که در اینجا برای forward کردن پیام ها استفاده شده به این صورت است که در خود پیام، مبدا و مقصد مشخص شده و هر گره دروازه با بررسی مقصد و با توجه به لیست خود آن را به گره مربوط ارسال می کند. برای اینکه پیام های اکتشافی بین خوشه ها به مقصد درست برسند گره های دروازه هنگام دریافت یک پیام اکتشافی از یک گره درونی به گره بیرونی، محتوای لیست ارسالی را به نحوی تغییر می دهند که آی پی های داخلی (خصوصی) به آی پی عمومی گره دروازه بدل گردد و از طرفی نوع پیام را از پیام اکتشاف ساده به پیام اکتشاف مرزی تبدیل می کنند و سپس به گره دروازه بعدی ارسال می کند، این گره دروازه نیز مشابه گره قبل همان کار را تکرار می کند و لیست را به دست گره مقصد می رساند. یک نمونه از تبادل پیام ها در شکل زیر آمده است.



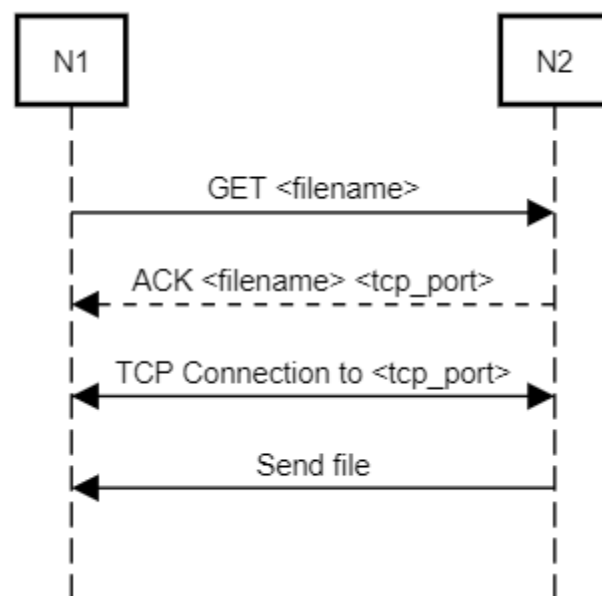
• فرایند تقاضا و انتخاب گره مقصد

زمانی که یک گره یک فایل را درخواست میکند، این درخواست به تمام گره های موجود در لیست آن گره ارسال می شود (به گره های هم خوشه به طور مستقیم و گره های خارجی از طریق گره دروازه و مکانیزم

(forwarding). گره مقصد در صورت داشتن فایل، پیام ACK را به همراه شماره پورت TCP خود به گره مبدا ارسال می کند و در غیر اینصورت پیام NACK ارسال می کند. گره درخواست دهنده از زمان ارسال درخواست فایل تا زمان دریافت پیام تایید یا رد گره ها را اندازه می گیرد و از گره با تاخیر کمتر دریافت می کند. برای این اندازه گیری، زمان ارسال درخواست در یک دیکشنری نام و زمان ارسال درخواست به گره ها ثبت میشود، از طرفی یک تایمر به اندازه حداکثر زمان انتظار می شمارد و در حین این شمارش، گره هایی که پیام ACK ارسال کنند زمان آن در دیکشنری ذخیره می شود و گره هایی که پیام NACK ارسال کنند از دیکشنری حذف می شوند، پس از اتمام زمان تایمر در بین گره های باقیمانده، پیمایش می شود و تاخیر بر اساس تفاضل زمان ارسال درخواست و دریافت تاییدیه محاسبه می شود و گره با کمترین تاخیر انتخاب می شود. عملیات انتخاب گره در یک نخ جداگانه peer_selector انجام می گیرد.

• فرایند انتقال فایل

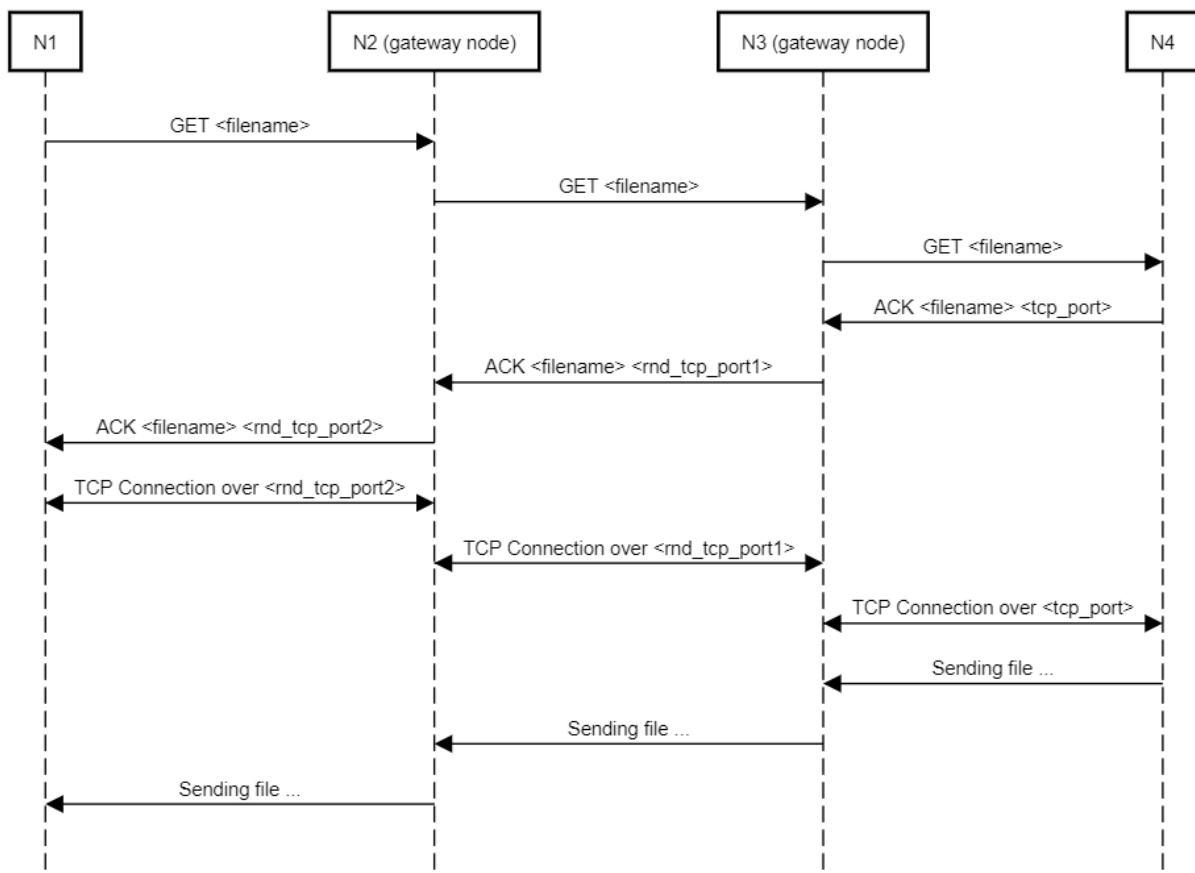
پس از تقاضای فایل و دریافت شماره پورت TCP مقصد در صورتی است که دو گره در یک خوشه باشند یک اتصال TCP به طور مستقیم از گره مبدا به گره مقصد شکل می گیرد و فایل منتقل می شود (در نخ جداگانه ی download_handler). شکل زیر این فرایند را نشان می دهد.



اما اگر این دو گره در دو خوشه متمایز باشند گره های دروازه به عنوان واسطه یا proxy عمل می کنند و در هر مرحله از forward پیام یک پورت TCP تصادفی را در نظر گرفته و با پورت TCP مقصد عوض می کنند تا به گره مبدا برسند. در این مرحله هر گره دروازه یک سوکت کلاینت TCP و یک سوکت سرور TCP با پورت تصادفی برای انتقال فایل اختصاص داده است و به عنوان سرور به درخواست گره بعدی گوش میدهد (یک نخ یا Thread مخصوص ایجاد میشود). گره مبدا با دریافت پورت TCP، در خوشه خود به گره دروازه با پورت تصادفی اعلام شده از طرف آن متصل می شود (گره دروازه از قبل به عنوان سرور TCP منتظر

درخواست دادن گره مبدا مانده بود)، این گره دروازه نیز با پورت تصادفی اعلام شده از طرف گره دروازه بعدی به آن متصل شده و آن گره نیز به پورت واقعی اعلام شده از طرف مبدا متصل خواهد شد. حال گره های دروازه به عنوان واسط از گرهی به گره دیگر داده ارسال و دریافت میکنند تا فایل منتقل گردد.

این فرایند در شکل زیر به تصویر کشیده شده است:



*گره های ۱ و ۲ در خوشه اول و گره های ۳ و ۴ در خوشه دوم قرار دارند. گره های ۲ و ۳ به ترتیب گره های دروازه ی خوشه های ۱ و ۲ هستند.

• مکانیزم توزیع بار

این مکانیزم مطابق فایل توضیح پروژه با محدود کردن تعداد گره های سرویس گیرنده این کار را انجام می دهد. این تعداد به صورت پیش فرض با عددی ثابت با نام DEFAULT_MAX_SERVICING_NODES مقدار دهی اولیه می شود. این مقدار قابل تنظیم بوده و از طریق سویچ msn- یا max_servicing_node -- قابل تنظیم است.

• مکانیزم کنترل سواری مجانی

این مکانیزم مطابق فایل توضیح پروژه با ایجاد تاخیر دستی از سواری مجانی جلوگیری می نماید. این مقدار تاخیر با ثابت MANUAL_DELAY_TO_RESPOND تعیین می شود. این مقدار قابل تنظیم نیست و به صورت Hard code وارد شده است.

• تنظیم و نحوه استفاده

سوییچ ها:

- ۱- سوییچ n یا name نام گره را تعیین می کند.
- ۲- با سوییچ l یا list نام فایل لیست خوشه را مشخص می کنیم.
- ۳- با سوییچ p یا port شماره پورت UDP را تعیین می کنیم.
- ۴- با سوییچ d یا directory پوشه ذخیره ی فایل ها را مشخص می کنیم.
- ۵- با سوییچ i یا discovery_interval فاصله زمانی بین ارسال هر دو پیام اکتشاف را معین می کنیم.
- ۶- با سوییچ mwt یا max_waiting_time حداکثر زمان انتظار برای دریافت پاسخ تاییدیه تقاضای فایل را مشخص می کنیم.
- ۷- با سوییچ msn یا max_servicing_nodes حداکثر تعداد گره هایی که میتوانند از گره فعلی فایل دریافت کنند را محدود می کنیم.
- ۹- با سوییچ g یا gateway که یک پرچم است مشخص می کنیم که آیا گره فعلی گره دروازه است یا خیر.

نمونه:

```
-python netwolf.py n N1 -p 8000 -l c1.txt -d n1 -g -i 5
```

این خط، گره دروازه ی N1 را با پورت 8000 UDP، فایل لیست c1.txt، پوشه ی n1 و فاصله ی زمانی اکتشاف ۵ ثانیه مشخص می کند.

نکته: زمان ایده آل برای اکتشاف برای ۵ ثانیه و بیش از آن مناسب و پایین تر از آن زیاده از حد است و ممکن است عملکرد گره های دیگر در forward یا پاسخ دهی به گره های دیگر را با کندی مواجه کند.

