گزارش تمرین کامپیوتر ی شبکه : Distance Vector Routing

هانیه ملائی

محمد موسوی

شرح ساختار کلی کد :

کد ما شامل ۴ کلاس مربوط router های ۰ تا ۳ است که با نام های Node_0 ...Node_3 نامگذاری شده اند و همگی از کلاس Node که کلاسی abstract و شامل تابع هایی نظیر ()rtinit و ()rtupdate و ()sendRouterPacket و تابع مربوط به چاپ distanceTable است که در هر node پیاده سازی میشوند .

از کلاس Main برای گرفتن ورودی ها از کاربر و فراخوانی سازنده ی کلاس NetworkSimulator و اجرای آن استفاده میشود .

و کلاس هایی برای event های مختلف و هندل کردن لیست مربوط به آنها گذاشته ایم .

شرح برخی تابع های مهم :

تابع ()rtinit در Node :

پس از ایجاد node ها در روند اجرای کد این تابع برای هر node فراخوانی میشود و با استفاده از آرایه ی دو بعدی اولیه ی تعریف شده که نشان دهنده ی هزینه های کلی همه ی مسیر های شبکه هست ، این تابع distanceTable هر node را مطابق هزینه ای که برای رسیدن به هر node همسایه ی خود دارد مقدار دهی میکند . البته در ابتدا همه ی فواصل distanceTable به صورت بینهایت یا همان ۹۹۹ در نظر گرفته شدند و در حلقه ی بعدی node های همسایه ی هر نود و هزینه ی بینشان مشخص شد .

در انتهای این تابع پاکت هایی برای اطلاع رسانی distanceVectore هر node به nodeهای همسایه اش که به شبکه افزوده شده اند و فعال هستند ، توسط تابع ()sendRoutePacket ارسال میشود .

تابع rtupdate(p Packet) در Node :

همان طور که در صورت پروژه نیز گفته شده بود این تابع برای زمانی پیاده سازی میشود که از node های دیگر پاکتی به node خاصی ارسال میشود که حامل node distanceTable دیگریست . این پاکت میتواند روی node distanceTable ماهم تاثیر بگذارد و راه کم هزینه تری برای رسیدن به یک node به ما معرفی کند که در این صورت باید به روز رسانی بشود . در صورتی که این اتفاق افتاد از این node به node های همسایه اش پاکت ارسال شود .

تابع sendRouterPacket() در Node :

این تابع distanceTable مربوط به node ای که درونش پیاده سازی شده را به شرط این که همسایه های node پیش از این init شده باشند و در شبکه حاضر باشند به همسایه ها میفرستد .

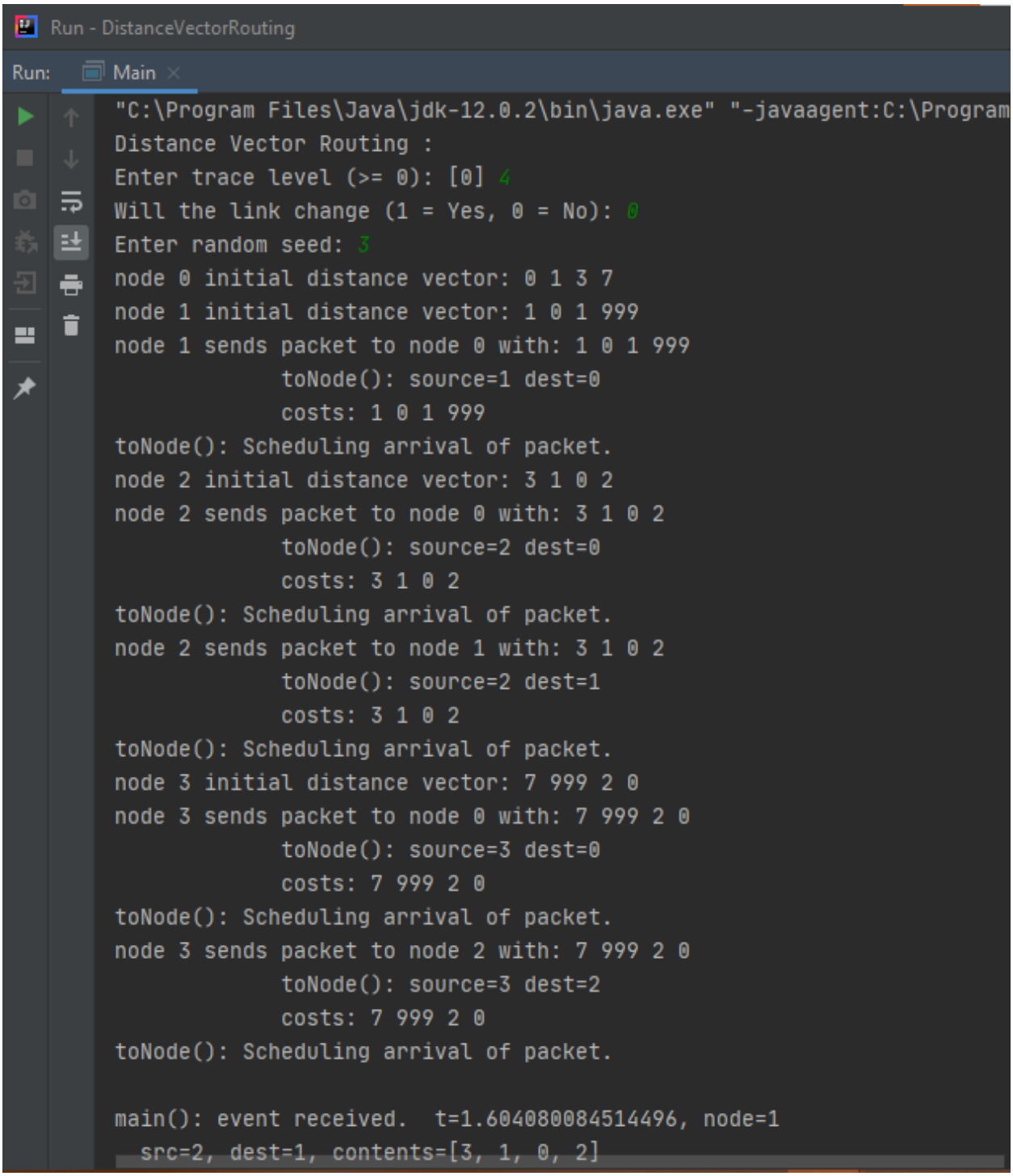
تابع printDT() در Node :

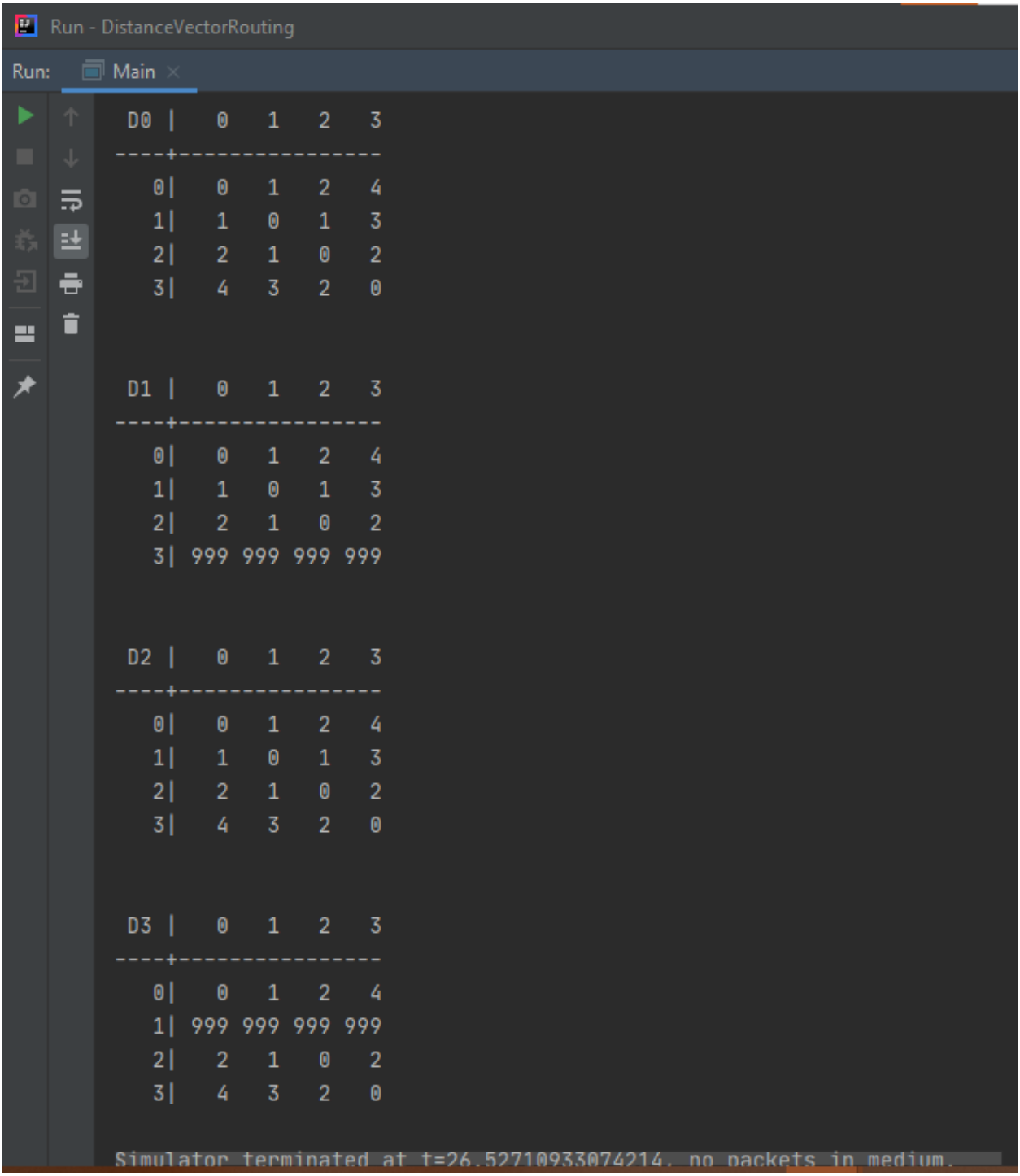
از این تابع برای چاپ distanceTable هر node استفاده میشود .

تابع toNode(p Packet) در NetworkSimulator :

این تابع صحت محتوای پاکت ها را بررسی میکند و خطاهای موجود را گزارش میکند و event ها را ایجاد میکند .

خروجی های مشاهده شده از کد :


Run - DistanceVectorRouting
Run:
Main
"C:\Program Files\Java\jdk-12.0.2\bin\java.exe" "-javaagent:C:\Program
Distance Vector Routing :
Enter trace level (>= 0): [0] 4
Will the link change (1 = Yes, 0 = No): 0
Enter random seed: 3
node 0 initial distance vector: 0 1 3 7
node 1 initial distance vector: 1 0 1 999
node 1 sends packet to node 0 with: 1 0 1 999
toNode(): source=1 dest=0
costs: 1 0 1 999
toNode(): Scheduling arrival of packet.
node 2 initial distance vector: 3 1 0 2
node 2 sends packet to node 0 with: 3 1 0 2
toNode(): source=2 dest=0
costs: 3 1 0 2
toNode(): Scheduling arrival of packet.
node 2 sends packet to node 1 with: 3 1 0 2
toNode(): source=2 dest=1
costs: 3 1 0 2
toNode(): Scheduling arrival of packet.
node 3 initial distance vector: 7 999 2 0
node 3 sends packet to node 0 with: 7 999 2 0
toNode(): source=3 dest=0
costs: 7 999 2 0
toNode(): Scheduling arrival of packet.
node 3 sends packet to node 2 with: 7 999 2 0
toNode(): source=3 dest=2
costs: 7 999 2 0
toNode(): Scheduling arrival of packet.
main(): event received. t=1.604080084514496, node=1
src=2, dest=1, contents=[3, 1, 0, 2]



همان طور که از خرجی هایی که به صورت جدول در انتهای اجا میآیند ، هر node اطلاعات مربوط به کوتاه ترین مسیر خود را ارائه میدهد . ولی ما کد را طوری تنظیم کردیم که متناسب با مفهوم distance vectore که فقط اطلاعات همسایه های خود را دارد با استفاده از پاکت هایی که همسایه ها به هم ارسال میکنند در نهایت همه ی اطلاعات هر روتر در خروجی مشخص شود .