Assignment 4: Take It Live

CS 356 UT Austin · Fall 2026 · Student handout

In A1 you made a network work. In A2 you made it fair. In A3 you made it visible. In A4 you run it for real, against everyone else. Your team is an Internet service provider. Your customers, homes and companies such as Netflix and Zoom, pay you for good service, and the rest of the Internet is the other teams. **The catch: you cannot do well alone.** You can always send your traffic through the Tier-1, a big provider we run that reaches everywhere. **The Tier-1 never fails, but it is slow and expensive.** The real value lies in connecting with the teams around you, and every connection is a deal you have to strike. The same neighbours compete with you for company contracts and carry your homes' traffic. This is **competitive cooperation**, and it is how the real Internet works: rival ISPs connect and exchange traffic because each needs the others to reach everyone.[1]

A4 is a game played live for five weeks. Two or three times a week (Figure 1) every team's network runs at once, in four worlds, on traffic nobody has seen. Between rounds you change your programs, bid for hardware, and negotiate agreements with other teams. Every agreement is a small program that decides who pays whom, based on what was actually measured. Everything is logged, and every round can be replayed. So you are working on three things: a network that carries traffic well, deals that make that traffic pay, and reading your replays to do better next round.

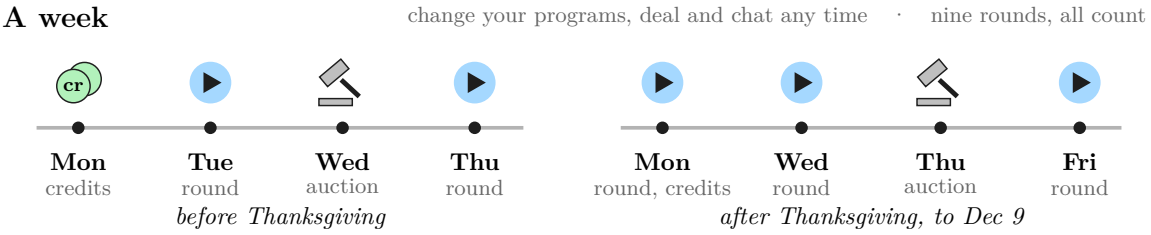


Figure 1: A4 at a glance: a typical week. Nine rounds, all counted, from Tuesday Nov 10 to Wednesday Dec 9.

The exact rules are in the **A4 rule book**, the last section of this handout (Section 12).

1 What you will learn

By the end of A4 you will be able to:

- **Routing policy.** Turn each agreement into policy: rank the routes you hear and choose the ones you announce. You choose how your traffic leaves, but others' traffic reaches you only along the routes you announce. Explain why the rule "prefer customers, and announce peer and provider routes only to customers" keeps routing stable with no one coordinating.
- **Routing as a distributed system.** Your routes are the outcome of every AS's policy, not just yours. Multi-home so that one failure cannot cut you off, follow a withdrawal as the network converges, spot routes that flap because policies conflict, and catch a leaked or broken announcement from your replay and the border meters.

[1] H. Balakrishnan, *Wide-area Internet routing*, MIT 6.033 lecture notes.

- **The economics of networks.** See why the real Internet looks the way it does: who peers, who pays for transit and where capacity gets built follow from money as much as from engineering. Every route you announce and every link you add shows up on a bill, and every deal you sign changes where traffic flows.

2 Background: how ISPs connect

The Internet is a network of networks. A network run by one organisation is an *autonomous system* (AS): an ISP, a university, a company such as Netflix. Inside its own AS an operator routes however it likes; that is what you did in A1–A3. Between ASes, routes are exchanged with BGP: an AS *announces* the address blocks it can reach to its neighbours, and each neighbour decides which announcements to use and which to pass on.

Who pays whom. Two ASes that connect make an agreement, and it decides both the money and the routes.

- **Transit**: a customer AS pays a provider AS to carry its traffic to (almost) everywhere. A small ISP buys transit from a bigger one.
- **Peering**: two ASes carry traffic between each other's customers, usually for free, because both gain. In *paid peering* the side that sends much more pays for the excess.

An ISP is paid by its own customers, pays its providers, and swaps traffic with its peers. *Example*: when a Comcast subscriber watches Netflix, the subscriber pays Comcast and Netflix pays its own provider (in 2014, Cogent). Neither network is paid by the other's customer; what Cogent and Comcast owe each other is a separate deal. In 2014 their link filled, Comcast would not upgrade it for free, Netflix streams slowed, and Netflix agreed to pay Comcast for a direct link (the inside story; Prof. Venkat Arun mentioned this example in class). A4 works the same way: your homes pay you. At an IXP, traffic flows between two teams only while they have an agreement, and carrying anyone else's traffic earns you nothing unless that agreement pays for it.

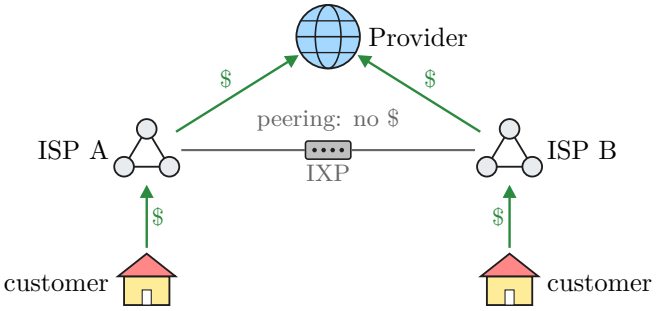


Figure 2: Money flows up: customers pay providers for transit. Peers swap traffic at an IXP without paying each other.

Announcing is a promise. If you announce a route to a neighbour, you offer to carry its traffic there, so ISPs choose their *export policy* carefully. The usual rule (Gao–Rexford): tell everyone about your customers, because carrying traffic to them earns money; tell your customers about everything; and do not pass routes learned from one peer or provider to another peer or provider, or you carry traffic between two other networks for free. A route announced by mistake (a *leak*) can pull in a flood of traffic nobody pays you for.

Where ASes meet. Mostly at *Internet exchange points* (IXPs): a shared switch in a city where many ASes connect and set up sessions with each other. Two ASes with a lot of traffic between them may also build a private link.

In A4. You are an ISP. The *Tier-1* is the big provider everyone already buys transit from: it reaches everything, and **it never fails, but it is slow and expensive**. Every other team is a possible customer, provider or peer, and you meet them at IXPs. The deal is an agreement (a settlement program); your routing policy (a policy program) ranks the routes you hear and chooses which you announce. For more, read the MIT 6.033 notes on wide-area routing.

3 Your network and your customers

The world. Eight IXPs form a ring (Figure 3), each in its own town. Between two neighbouring IXPs is a *sector* of four ASes. Each AS has a hub in both towns of its sector and a port at both IXPs, so it can meet 11 others: 3 in its sector and 4 in each neighbouring sector. Each AS also has a link to the Tier-1 in each town; **the Tier-1 never fails, but it is slow and expensive**. Nothing crosses the middle of the ring except microwave links sold at auction, so traffic to the far side (the five sectors beyond your neighbours) goes around the ring or over the Tier-1. You play four worlds on the same map, with different neighbours in each.

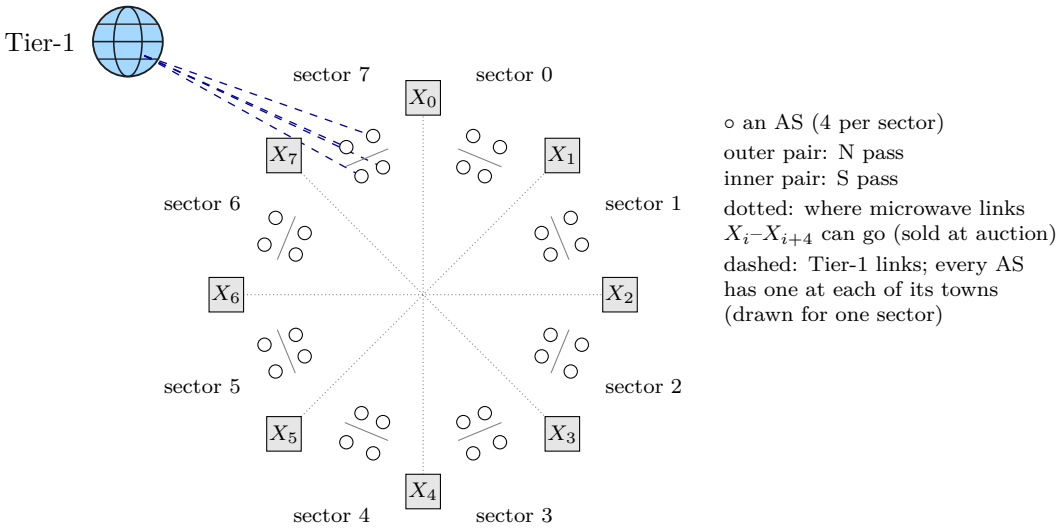


Figure 3: One world. Every AS attaches to the IXPs at both ends of its sector and to the Tier-1 in each of its towns.

Passes. A ridge separates the two towns of a sector (Figure 4). It has two passes, N and S. Each AS's own link between its two towns, its *crossing*, runs through one of them, and two ASes use each pass. A landslide in a pass cuts both crossings at once, so your pass-mate cannot be your backup; the two ASes in the other pass can.

Your AS. Seven switches, the same for everyone (Figure 5): a hub in each town (W and E) with your IXP port and your Tier-1 link; two access switches under each hub, where your homes attach; a hosting site H, where the companies attach; and the *crossing* between W and E through your

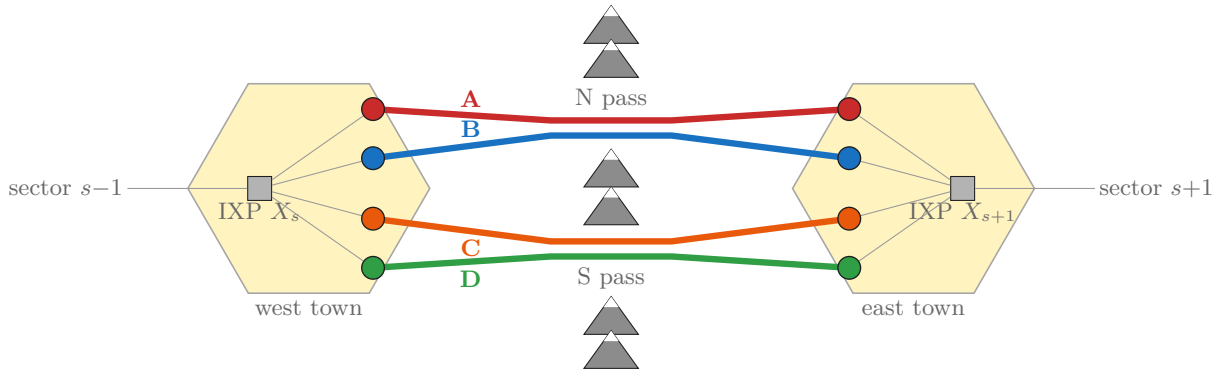


Figure 4: One sector. Each AS has a colour and a hub (dot) in each town, at that town's IXP. Its crossing (road) runs over the ridge through a pass: A and B use the N pass, C and D the S pass. A landslide in the N pass cuts A and B at once; C and D are unaffected.

pass. If the crossing fails, your two halves can reach each other only through another AS or the Tier-1.

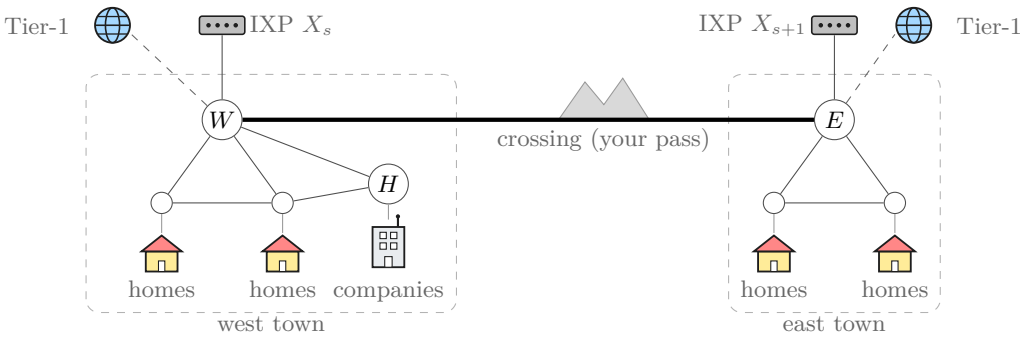


Figure 5: Your AS, the same for every team. Each town has a hub (W, E) with your IXP port and your Tier-1 link, and two access switches for homes; H, in the west town, hosts the companies, so they reach your east-town homes over the crossing. The crossing is your only link between the towns.

Your customers. Two kinds, all in your own sector.

- **Homes.** Each AS has its own homes, the same number for everyone, in both its towns. They stay with you all game, and pay a price per Mb (megabit) that arrives on time, for traffic they send and receive. (Real homes pay a flat monthly fee and leave when service is bad; paying per Mb on time puts that into every round.)
- **Companies.** Each sector has one site each of Amazon, Netflix, Zoom and Jane Street, connected to all four ASes. Each sends a quarter of its traffic through each AS and splits a fixed contract among the four by how much of that traffic each delivered on time, weighted so that the best gets more.

Company money is competitive. Each contract is a fixed amount, so what one sector-mate wins, the others lose. **Home money is cooperative.** You are paid only when your homes' traffic arrives on time, and other teams carry much of it: three quarters of what your sector's companies send your homes comes in through a sector-mate, and traffic to the rest of the ring crosses the ASes

beyond. The same three neighbours are your rivals for the first kind of money and your partners for the second.

The traffic. About a quarter each goes within your sector, to the sectors next to it, to the far side of the ring (the other five sectors), and to content reached only through the Tier-1. Each company is one class with its own latency target: Web (Amazon) and Video (Netflix) arrive on time even over the Tier-1 while it has room, Calls (Zoom) only sometimes, and Trading (Jane Street, between its own sites) never. The A4 rule book (Section 12) has the details; `rules.json` has the numbers.

4 The decisions you make each round

The MIT notes split an ISP's choices in two. First, *whom you connect with, and on what business terms*: peering or transit. Second, your *routing policy*, which turns those deals into routes: *filtering* (which routes you accept, and which you announce to whom) and *ranking* (which of the routes you accept you prefer). In A4 we do the import filtering for you (you are offered only routes your agreements cover), and your policy program ranks and announces; these are the two big decisions, but not the only ones. Between every pair of rounds, in each of your four worlds, you decide (Figure 6):

1. **Agreements: peer or transit, with whom, and on what terms.** For each neighbour: no agreement, *peering* (each carries traffic to the other's customers, no money changes hands), *transit* (one pays the other to reach everyone), or a variant such as paid peering or backup transit. Then the terms: which template, its prices, whether to add a refund, and how many rounds it runs. For agreements that start at the Round 2 cutoff or later, you may write your own settlement program instead. One team offers, the other signs. It runs for the rounds you name and then renews unless either side declines; neither can leave early alone. Every agreement, payment and bid is in a public *ledger* that all teams see.
2. **Routing policy: rank, then announce.** *Rank*: among the routes you are offered for a destination, which you use; this decides where your traffic goes and what you pay. *Announce*: which of those routes you tell each neighbour about; this decides whose traffic you carry. If you change neither, you get the usual rule: prefer any route through another team over the Tier-1, then the shortest path, then customers (here, ASes that buy transit from you) over peers over providers; announce your own and your customers' routes to everyone, and your peers' and providers' routes only to your customers. Announcing more is a *leak*: it pulls in traffic nobody pays you for.
3. **Your network.** Where traffic leaves your AS, which classes your queues serve first, and how you notice that your crossing is down and fail over.
4. **Hardware.** In auction weeks, what to bid for, alone or with other teams; at any time, whether to trade hardware with another team, and where to install what you own.
5. **When and how you change.** You may resubmit your programs, sign new agreements and place bids as often as you like before a cutoff, on the website.

Then you submit and sign before the cutoff, we run the round, and each agreement's settlement program turns our meter readings into payments. The statement you get back starts the next round's decisions.

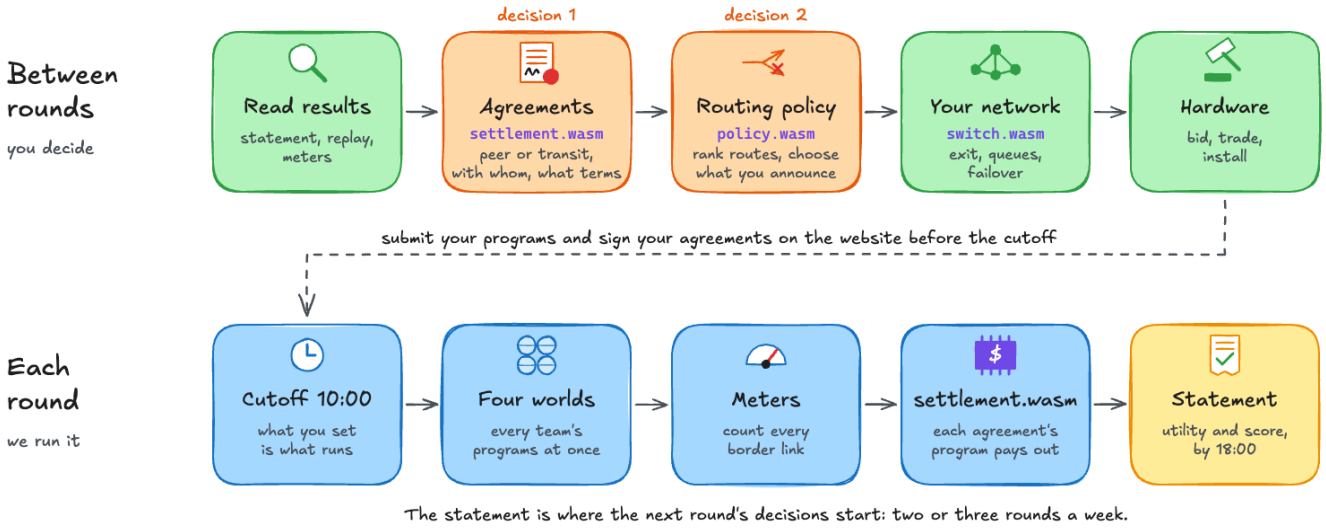


Figure 6: Your round. Above: what you decide between rounds; the two orange boxes are the decisions from the MIT notes, and the purple names are the three programs you submit. Below: what we run with what you set, including the settlement program behind each agreement.

5 What you build

- **`switch.wasm`, your network.** One program on every switch of your AS, as in A1–A3. It makes three decisions:
  - *Exit*: which town each destination leaves from. Hand traffic off at your nearest exit, or carry it across your crossing to the exit nearest its destination.
  - *Queueing*: at your borders and on your crossing. Calls and Trading have tight latency targets; Web and Video can wait.
  - *Failover*: when your crossing fails, reconnect your two halves through a neighbour or the Tier-1.
- **`policy.wasm`, your routing policy.** Two small functions we call when we work out the round's routes: *rank* (which offered route you use for each destination) and *announce* (whether you tell a neighbour about a route). Your switch program then forwards along the routes you chose. Start from our policy program, which follows the usual rule; if you submit none, the usual rule is what you get. Announcing a route is a promise to carry the traffic, so your policy must match your agreements.
- **Hardware auctions.** Every team starts with the same network. From the first auction you can bid for five kinds of hardware with *credits* (cr): 100 cr per world when the game opens and every Monday after, for hardware only. Credits are not dollars and not part of your score.
  - a *second cable* in your pass: more capacity, same risk;
  - a *microwave* link: low latency across the ring, to a switch of yours at the IXP on the far side;
  - a *private interconnect*: your own link to one neighbour, beside the shared IXP (Netflix and Comcast's 2014 deal was one);
  - *upgrades*: more capacity on links you own, bigger buffers, faster ports, more program memory;

🔧 a *repair crew*: your links come back sooner after a failure.

A separate hardware handout, out with the first auction's lots, has the details: how auctions work, what each item does, and how to install it. Links this costly are often built together: most undersea cables are owned by groups of operators (TeleGeography's cable FAQ), and trading firms built their own microwave networks between Chicago and New Jersey to save milliseconds (Bhattacherjee et al., IMC 2020). You may bid jointly with other teams.

- **Settlement programs (`settlement.wasm`), your agreements.** Each agreement with another team is a small wasm program that reads our meters after a round and returns who pays whom. Start from our templates (transit, peering, paid peering, backup transit) and fill in the numbers, or, from Round 2, write your own terms.
- **Your strategy.** Whom to connect with and on what terms, what hardware to bid for and where to put it, when to resubmit. You do this on the website; everything you submit is a wasm program.

Why you keep changing it. Your network does not run alone. If a peer drops you, traffic that used to leave from one town now crosses to the other and loads your crossing. A transit customer you sign adds traffic that crosses your AS. A failed pass shows whether a repair crew or a backup agreement is worth its price. Each replay shows which of these happened; your final report says what you changed after each round.

6 Your first week

The first round is Round 1, on Tuesday Nov 10, and every round counts. Week one has two steps; their dates are in the schedule at the end of this handout.

1. Your own network. Get the kit from Canvas and run the starter program on the practice world. Until you sign agreements, every AS reaches the others only through the Tier-1, so start with the three decisions that are yours alone:

- *Exit*: hand traffic to the Tier-1 at the town where it enters your AS, or carry it across your crossing and hand it off at the other town.
- *Queueing*: what goes first at your borders and on your crossing. Calls and Trading have tight latency targets; Web and Video can wait.
- *Failover*: when your crossing fails, reach your other town through the Tier-1 or a neighbour. Test this with the practice world's landslide.

2. The game opens. The game website opens with a *website guide*, and the numbers in `rules.json` freeze. Log in, choose your team's *pseudonym* (other teams see it, never your names; if you do not choose one, we give you a random one), find your seat (your AS's place on the map) in each of the four worlds, and talk to your neighbours. Submit your switch and policy programs there: every submission gets a dry run on the practice world, so you see within minutes whether it runs and what it earns there. Submitting the starter unchanged is fine. Sign at least one peering agreement from a template before the Round 1 cutoff; our starter and policy programs use the routes your agreements bring, so it pays off without a new program. This is advice, not graded. The hardware handout and the first auction's lots (the items for sale) come out the same day. After Round 1, read your replay and statement and note what you learnt: your final report is built from these notes.

7 What you get

Everything from A1–A3: the simulator, the Rust SDK, the replay viewer. New in this kit:

- **The practice world**: a small ring of exchange points with a Tier-1, meters, companies and homes, and other ASes running the starter program. Run it as often as you like, with your own program in any seat, and add an agreement to see what it pays before you sign it.
- **A starter switch program** that forwards along the routes your policy chooses (with no agreements, the Tier-1), and **a policy program** that follows the usual rule.
- **Settlement templates** as source.
- **The API guide**: what your switch program is told, the policy program's two functions, and the settlement program interface.
- `rules.json`: every number in the rule book, for your programs. Its values may change until the numbers freeze (see the schedule); the current copy is always on the website.

8 How your score works

Every world and round gives you a *utility*, in dollars:

$$\text{utility} = \text{value} + \text{what other teams pay you} - \text{what you pay them} - \text{the Tier-1 fee}.$$

- **Value** is what your own customers pay you: your homes pay for their traffic that arrives on time, and each company pays you a share of its contract that grows with how much of its traffic you deliver on time.
- **What teams pay each other** is set by your agreements.
- **The Tier-1 fee** is charged for every Mb on your Tier-1 links.

Raw dollars depend on the round's traffic and failures, so your *score* compares your utility with two runs we make on the same world, traffic and failures: the *floor*, where every AS runs our reference program and uses only the Tier-1 (score 0), and *par*, where every AS runs our reference program with our reference agreements (score 100):

$$\text{score} = 100 \times \frac{\text{your utility} - \text{floor}}{\text{par} - \text{floor}}.$$

Above 100 is better than our reference; below 0 is worse than using only the Tier-1. Your *season score* adds up your score in every world and every round: four worlds times nine rounds, 36 scores in all. The leaderboard ranks teams by season score. The prices, contracts and fees are in `rules.json`, and each round's statement shows every line of your utility.

9 Grading

You are graded on two things, and the final report is the only thing you hand in.

- **Season score, 75%.** Graded from the leaderboard after Round 9, the last, with a scoring function we will announce later on Ed. You hand in nothing for it. At each cutoff we run the last program you submitted that passed its dry run, so if you submit nothing new, that one keeps running (our starter and policy programs if you never submit). Agreements, bids and their settlement programs are made on the website, and count as they stand at each cutoff.

- **Final report, 25%**, 5 pages, on Canvas by Friday Dec 11, 23:59. Round by round, from your replays and statements: what you decided and why, what the round taught you, and how you changed your strategy for the next round. It is graded on evidence and reasoning, never on your score: a team that lost money and explains exactly why can earn full marks. Write a few notes after every round; nine rounds are hard to reconstruct from memory.

Quiz 3 (Tuesday Dec 1) asks about your own replays from Rounds 1–5.

10 Using AI

Use it however you like, including an AI agent that writes your programs and helps you plan. There is no game API: you play on the website, and everything you submit is a wasm program (switch, policy and settlement programs), which your agent can write; do not script the website. AI knows how the Internet's providers connect; it has never seen these worlds or your neighbours. The deals you make are with people, and with four AI-played ASes (marked on the map), who are also trying to win.

11 Logistics

Rounds	Nine rounds, all counted, and three auctions, dated in the schedule below.
Teams	2 students: your A3 team. Teams are fixed for the whole game.
Help	Questions on the course discussion board; game bugs as in the website guide; office hours as usual.
Rules	The A4 rule book (Section 12); its numbers are in `rules.json`.

Schedule. Times are Central Time. A round's time is its cutoff: what you have set then is what runs. Each round's results are out by 18:00 the same day.

Date	Time	What happens
Thu Nov 5		Handout out
Fri Nov 6	18:00	Kit out on Canvas
Sat Nov 7	10:00	Game opens; numbers freeze; hardware handout and first lots out
Tue Nov 10	10:00	**Round 1**
Wed Nov 11	20:00	**Auction 1** closes
Thu Nov 12	10:00	**Round 2**
Tue Nov 17	10:00	**Round 3**
Wed Nov 18	20:00	**Auction 2** closes
Thu Nov 19	10:00	**Round 4**
Nov 23–27		Thanksgiving: no rounds
Mon Nov 30	10:00	**Round 5**
Wed Dec 2	10:00	**Round 6**
Thu Dec 3	20:00	**Auction 3** closes
Fri Dec 4	10:00	**Round 7**
Mon Dec 7	10:00	**Round 8**
Wed Dec 9	10:00	**Round 9**, the last; final leaderboard
Fri Dec 11	23:59	Final report due

12 A4 rule book

This section is the A4 rule book: the whole set of rules. If anything else (the website, an email, a TA, another part of this handout) disagrees with it, the rule book wins. It is written for you and for your AI agent. **Give it to your agent as standing context**: save the text of this section, with `rules.json`, where your agent's instructions file (for example `AGENTS.md` or `CLAUDE.md`) points, so it is loaded at the start of every session. Each rule stands alone; terms are defined in R1; every number is named by its key in `rules.json`, written as `key`.

Terms

R1. Definitions. *AS*: one team's network in one world. *World*: one copy of the game map (`world.worlds` of them). *IXP*: an exchange point; `world.ixps` form a ring. *Sector*: the `world.ases_per_sector` ASes between two neighbouring IXPs. *Hub*: your switch in each of your sector's two towns, with your IXP port and your Tier-1 link. *Crossing*: your link between your two hubs; it runs through one of the sector's two *passes*, shared with one other AS (your *pass-mate*). *Seat*: an AS's place on the map. *Round*: one run of all four worlds at a *cutoff*. *Utility*: your dollars in one world and round (R8). *Agreement*: a signed settlement program between two ASes. *Usual rule*: the default routing policy (R22). *Ledger*: the public record of every agreement, payment, bid and hardware sale. *Mb*: megabit.

The world

R2. Seats. Each world has `world.ases_per_world` ASes: the 28 teams of this class and `world.ai_seats` ASes played by AI, marked AI on the map. AI seats deal only through agreement offers (no chat) and act after each round's results and 8:00–9:00 CT each morning. Every seat is equivalent. The worlds are fixed for the game; you share a sector with any other team in at most one world.

R3. Neighbours. You can interconnect with the `world.possible_neighbours` other ASes at your two IXPs: 3 in your sector and 4 in each neighbouring sector. Traffic between two ASes at an IXP flows only while they have an active agreement. Only microwave links cross the middle of the ring.

R4. The Tier-1 has a link to each of your hubs and reaches every destination, including external content nothing else reaches. It never fails; its capacity is fixed and cannot be bought. We announce your own address blocks to it on both links, each as nearer the link it is closer to; you do not set this.

R5. Your network is the same for every AS: two hubs, two access switches under each (your homes), a hosting site in the west town (the companies) and your crossing. Capacities and latencies: `network`; buffers: `buffer_bytes_per_port`.

R6. What you can see. The website shows every world's map and the hardware on it, never live traffic or other ASes' routes. You see what your own switches measure and log, your statements, the meter readings on your own border links (every `meter_epoch_s`), and your round's replay: your own AS in full, other ASes only as what crossed your borders.

Money and score

R7. Two kinds of money never mix: *dollars* are your score; *credits* (cr) buy hardware.

R8. Utility in each world and round = value + settlements received − settlements paid − Tier-1 fee. *Value* is what your homes and companies pay you (R9, R10).

R9. Homes pay `classes[].home_price_per_mb` for each Mb, sent or received, that arrives within its class's `classes[].target_ms`; late or lost traffic pays nothing. When your home talks to a company through another AS, your home pays you.

R10. Companies each pay `classes[].company_contract_per_sector` per round, split among the sector's four ASes. If q_A is the fraction of the company's traffic through AS A that arrived within the target (both directions), A earns contract $\times\, q_A^2/(q_A^2 + q_B^2 + q_C^2 + q_D^2)$; the whole contract is always paid. *Example*: B, C, D deliver 95%, A 60%: A earns $0.36/3.07 \approx 12\%$, the others about 29% each.

R11. The Tier-1 fee is `tier1_fee_per_mb` for every Mb on your Tier-1 links, in each direction.

R12. Score. In each world and round, score $= 100 \times$ (utility − floor)/(par − floor), where *floor* is the run in which every AS runs our reference program and uses only the Tier-1, and *par* the run in which every AS runs our reference program with our reference agreements, on the same world, traffic and failures. Your *season score* is the sum over the four worlds and all `scoring.rounds_counted` rounds.

R13. Grade: season score 75%, graded from the leaderboard after the last round with a scoring function announced on Ed; final report 25% (`scoring.grade_weights`).

R14. Credits are separate in each world: `credits.at_open` when the game opens and `credits.per_monday` every Monday, spent in auctions and on hardware bought from other teams; unspent credits carry over up to `credits.carry_max`; they are worth nothing at the end.

Agreements

R15. Only agreements bind. Chat is logged and binds no one.

R16. A settlement program runs after each round on the readings for the links between the two ASes (per direction, class and `meter_epoch_s`: bytes crossed; how many arrived on time end to end, late or lost; and how many came straight back into the AS they left, which is one AS carrying the other's traffic between its own towns) and the round's failure log. It returns one amount and who pays it, never more than the maximum both teams signed. It can do nothing else; limits: `settlement_program`. If it traps or runs out of fuel, it pays nothing that round and the ledger records it.

R17. Templates. *Transit*: the customer pays the provider a price per Mb crossing between them, in either direction. *Peering*: no payment. *Paid peering*: if one side sends more than an agreed multiple of what it receives, it pays a price per Mb for the excess (what it sent minus that multiple of what it received). *Backup transit*: a fixed fee each round plus a price per Mb of the buyer's traffic carried between the buyer's towns. Transit and backup may add a refund, off unless set: an amount paid back if the partner does not carry the traffic when promised, or delivers too little of it on time.

R18. Your own settlement program may replace a template for agreements starting at the Round 2 cutoff or later. It is run on test readings when uploaded; one that traps or runs out of fuel is rejected.

R19. Signing. One team offers (a filled-in template or its own program) and names the world and the rounds it covers; the other signs. An agreement is identified by the hash of its program and its numbers, so what both sign is exactly what runs. It starts at the next cutoff and runs for the rounds named. It ends early only if both sign a cancellation, or after a rule change (R31). At the end of its term it renews for the same number of rounds unless either side declines before the term's last cutoff. An offer can go to the same seat relative to you in every world; each copy is signed separately. Two teams may hold more than one agreement in a world. Each agreement has a label (provider, customer, peer, backup) that your programs see.

R20. Metering. At an IXP, the exchange (run by us) counts the traffic between each pair of members; every other link that leaves your AS carries a meter run by us. Settlements and customer payments use only our measurements.

Your programs and routing

R21. Three programs. `switch.wasm` runs on every switch of your AS (exit, queueing, failover; limits: `switch_program`). `policy.wasm` is your routing policy (R22; limits: `policy_program`). Each agreement is a settlement program. All are wasm; the API guide has the interfaces.

R22. Routing policy. For each destination, your policy program chooses which offered route you use (*rank*) and whether you tell each neighbour about it (*announce*). You are offered only routes your agreements cover. Without a policy program, or if it traps, runs out of fuel or answers out of range, the *usual rule* applies: rank any route through another team over the Tier-1, then the shortest AS path, then customer over peer over provider; announce your own and your customers' routes to everyone, and your peers' and providers' routes only to your customers.

R23. Announcing is a promise to carry the traffic. Announcing more than the usual rule is a *leak*, and you carry that traffic unpaid.

R24. Settling. Routes are recomputed until no AS changes, at most `policy_program.max_iterations` times; if they have not settled, the last iteration is used and your statement says so.

R25. Backups are handled by us: the seller's route between your towns is offered to your switch program for failover; policies do not rank or announce it.

Rounds

R26. Schedule. The game opens at `schedule.game_opens`; round cutoffs are `schedule.round_cutoffs`, auctions close at `schedule.auctions_close`, all Central Time (the schedule in Section 11 lists them). Results are out by 18:00 on the day of the cutoff. Every round counts.

R27. The cutoff. What you have set at a cutoff (programs, installed hardware, signed agreements) is what runs; nothing changes during a round.

R28. Submissions. Every switch or policy program must pass a dry run on the practice world before it is accepted; resubmitting is free and has no limit. If your switch program fails during a round, that round is run with your last version that passed; if you never submitted one, with our starter.

R29. Traffic and failures. Each round runs on traffic you have not seen, the same in every repeat of the round. Single links, IXP ports and whole passes fail, on one schedule in which every sector and every pass is hit equally often over the season. Your programs and settings are never touched.

R30. Results. For each world: your utility and score, a statement for each agreement and each customer (the measurements it was paid on), your logs and the replay. The leaderboard shows season scores and your place in each world.

R31. Changes. Numbers in `rules.json` may change until the game opens. After that they change only to close an exploit or fix a broken rule, announced a week ahead, and any agreement the change affects may then be cancelled by either side.

Hardware and conduct

R32. Hardware is sold only in the auctions, for credits. Teams may bid jointly and trade hardware at any price both agree on. The hardware handout has the auction rules.

R33. Cooperation is allowed; sham deals are not. Deals that only move money between teams can be reversed.

R34. AI is welcome: it may write your programs and help you plan. There is no game API; do not script the website. Do not share your programs, or reach the servers or another team's AS except through the game.