Hello!

My name is Karl Auer

I'm with Into6

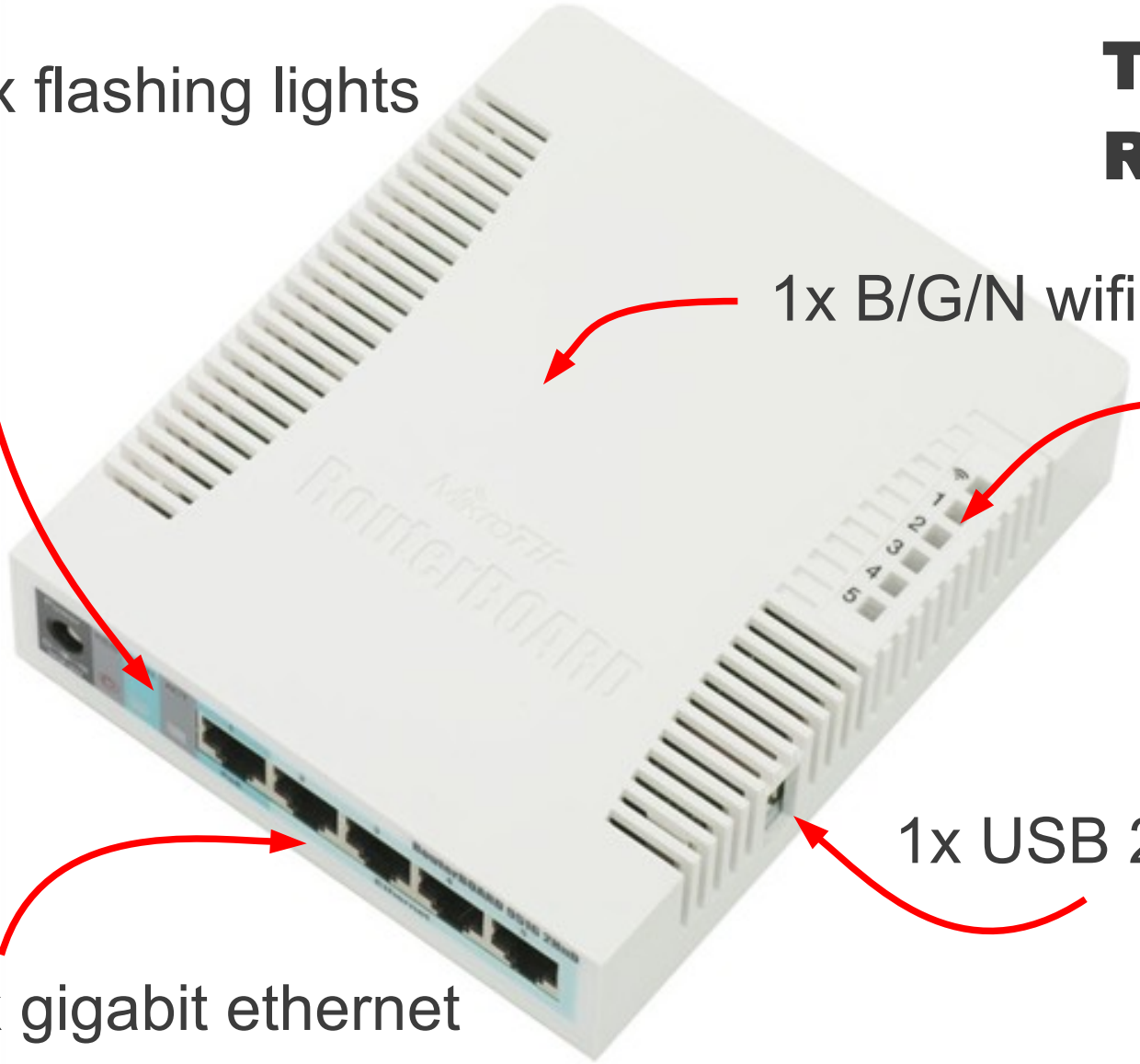
But today is about MikroTik

- and about networking...

- Latvian company; www.mikrotik.com
- Started in 1995, manufacturer since 2002
- Specialises in routing and wireless
- RouterBOARD hardware
- RouterOS software

A typicalMikroTik router

The MikroTik RB915G-2HnD



2x flashing lights

1x B/G/N wifi

6x flashing lights

1x USB 2.0

5x gigabit ethernet

Very rich feature set

- Very flexible port arrangements
- Flexible wireless (AP, station, bridge..)
- Excellent routing (OSFP, BGP, RIP...)
- Excellent tunnels (GRE, 6to4, EoIP, IPinIP...)
- IPSec, PPTP, L2TP support
- VLAN support
- IPv4 and IPv6 support

Very rich feature set

- Access via web, ssh, telnet, serial...
- Powerful firewall with scheduling
- Powerful NAT features
- Scripting
- Scheduling
- Logging to memory, file, network
- Can be powered by PoE

Very cheap...

- $100 delivered (RB951-2n is $50 delivered)
- Great home router (no ADSL though)
- Great for hobbyists, geeks
- Great for small business
- Cheap enough to play with
- Solid enough for the real world
- From eg www.duxtel.com.au

Some ideas...

- Use virtual APs for guest/gaming/kids wifi
- Separate your home and business networks
- Plug in a 3G dongle as a backup link
- Attach a disk or two – cheap NAS!
- Use one as an SMS gateway
- [… your brilliant idea here...]

Downsides

- No two-step configuration
- No easy cloning
- No HA features
- No power backup
- Plastic (good build quality though)
- Throughput is middle-of-the road
- Not enterprise grade

Playtime

- Connect via WiFi (SSID: `MikroTik`)
- Browse to `192.168.88.1`
- No password needed (yet)
- Look around!

Playtime

- Open a terminal window ("DOS box")
- ssh to `admin@192.168.88.1`

  OR
- telnet to `192.168.88.1`, login as `admin`
- No password needed (yet)
- Look around!

Initial config (1)

- Set the router identity
- Set the router password to “mikrotik”
- Set the Wifi SSID
- Set up WPA/WPA2 + passphrase
- DON'T click “Apply Configuration” yet!
- … more on next slide...

Initial config (2)

- Set the local LAN address if desired
  - "LAN IP Address" → 192.168.77.1/24
- Check "DHCP server"
- Set "DHCP Server Range" to match LAN
- Check "NAT" if appropriate
- DON'T click "Apply Configuration" yet!
- … more on next slide...

Initial config (3)

- Set the WAN interface appropriately
- Here we use “DHCP”
- But could use PPPoE with an ADSL modem
- Try the other options, see what changes
- Click “Apply configuration”
- Reconnect, log in
- You should have Internet

Recap

- The initial config is all you need for home
- All you need for a typical single-subnet site

  (You should configure a firewall though)

- But doesn't scratch the surface!

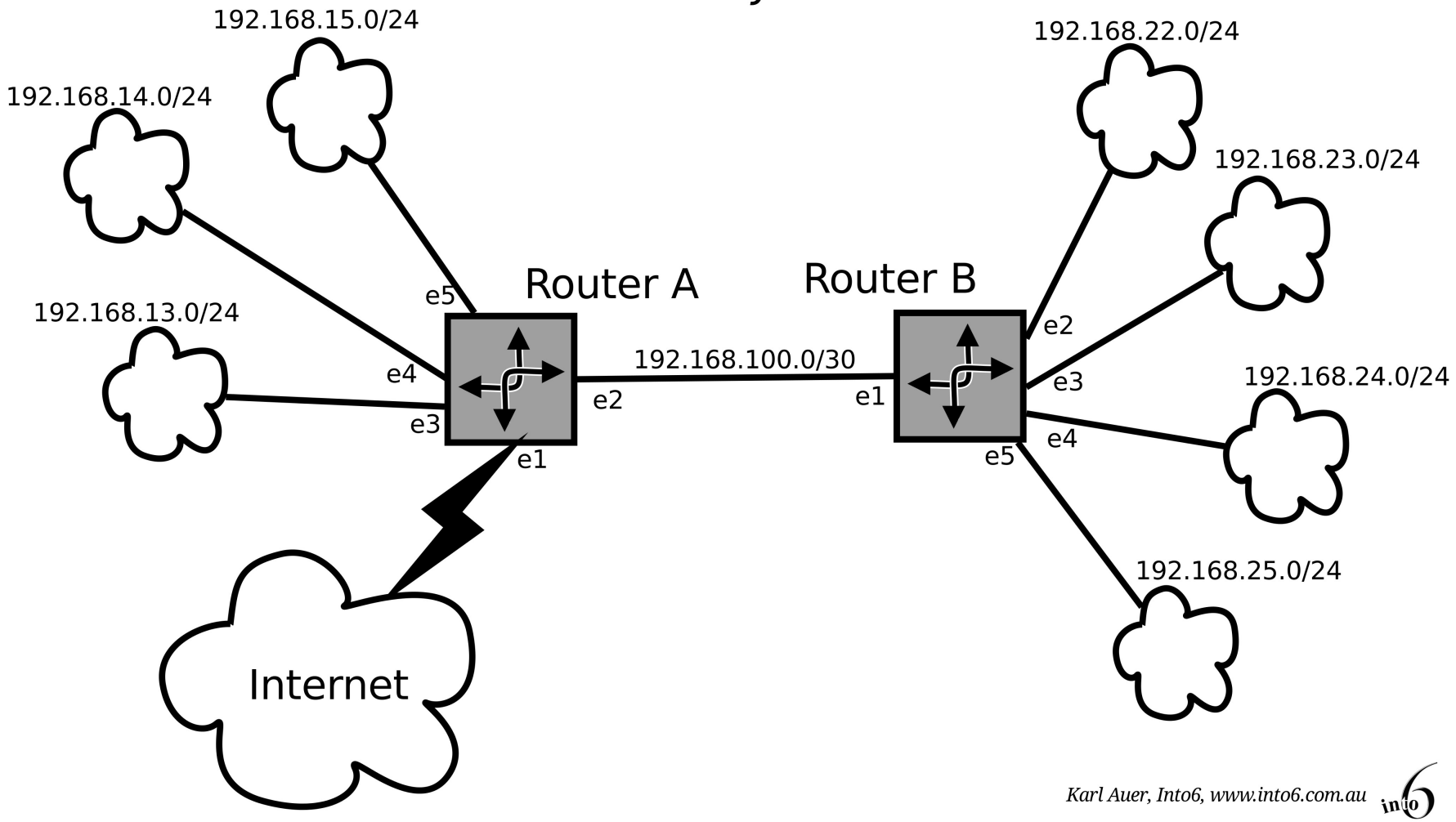
  … so let's build a *real* network.

Demo network - goal

- Two routers connected to one upstream
- Each router with a loopback address
- One subnet for each person
- All subnets can reach each other
- All subnets can reach the Internet
- Basic routing with OSPF
- Basic firewalling

MikroTik Demonstration Network

IntoIT 15 June 2013



Karl Auer, Into6, www.into6.com.au

Physical connections

- Disconnect from Internet!
- `ether1` on Router A will be Internet
- Router B will be an internal router
- `ether1` on Router B to `ether2` on Router A
- Remaining ethernet interfaces to laptops

Router A Internet interface (1)

- Set WAN to "DHCP"
- Check that it is NATting
- Set up firewall on gateway interface:
  - Allow established in/forward
  - Allow related in/forward
  - Allow ICMP in/forward
  - Block all else in/forward
- Reconnect Internet

Router A Internet interface (2)

- Check connectivity

```
/ping www.google.com
```

Router A downstream interface (1)

- Use `ether2`
- Take `ether2` out of the switch group

```
/interface ethernet
print detail
set X master-port=none \
    name=ether2
```

("set" is all on one line)

Router A downstream interface (2)

- Take `ether2` out of the bridge group

```
/interface bridge port
print detail
remove X
```

Router A downstream interface (3)

- Select a link subnet and interface address, e.g.:
- Link subnet: `192.168.100.0/30`
- Local address: `192.168.100.1`
- Remote address: `192.168.100.2`

Router A downstream interface (4)

- Put the local address on `ether2`:

```
/ip address
add address=192.168.100.1/30 \
    interface=ether2
```

(the “add” is all on one line)

Router B Upstream (1)

- Set WAN to “Static”
- Set the interface address to `192.168.100.2/30`
- Set the gateway address to `192.168.100.1`
- Turn off NAT

Router B Upstream (2)

- Add a default route:

```
/ip route
add dst-address=0.0.0.0/0 \
    gateway=192.168.100.1
```

(the “add” is all on one line)

Router B Upstream (3)

- Add a DNS server:

```
/ip dns
set servers=192.168.100.1
```

Router B Upstream (4)

- Check connectivity:

  ```
  /ping 192.168.100.1
  ```

- Check DNS

  ```
  /ping www.google.com
  ```

Everyone gets a subnet

- Step1: Pick an interface (check diagram)
- Step 2: Pick a subnet (check diagram)
- Step 3: Configure the interface
- Step 4: See if you can ping the interface from your laptop
- Step 5: What works, what doesn't?

New subnet on interface (1)

- Take your ethernet interface out of its switch group if necessary

```
/interface ethernet
print detail
set X master-port=none name=whatever
```

New subnet on interface (2)

- Take your ethernet interface out of bridge group if necessary

```
/interface bridge port
print detail
remove X
```

New subnet on interface (3)

- Select a subnet and gateway address for the interface:

  Subnet: `192.168.7.0/24`

  Gateway: `192.168.7.1`

- Use your selected subnet, not this!
- Make sure your subnet is unique!

New subnet on interface (4)

- Add the gateway address to your ethernet interface:

```
/ip address
add address=192.168.7.1/24 \
    interface=name_of_interface
```

(the “add” is all on one line)

New subnet on interface (5)

- Set up an address pool for your subnet:

```
/ip pool
add name=pool-fred
    ranges=192.168.7.10-192.168.7.100
```

(the “add” is all on one line)

New subnet on interface (6)

- Add a DHCP server to your interface:

```
/ip dhcp-server
add name=dhcp-fred
    interface=name_of_interface \
    address-pool=pool-fred \
    authoritative=yes
```

(the “add” is all on one line)

New subnet on interface (7)

- Tell the new server about the new subnet:

```
/ip dhcp-server network
add address=192.168.7.0/24 \
    gateway=192.168.7.1 \
    dns-server=192.168.7.1 (etc)
```

(the “add” is all on one line)

Check new subnet

- Does the router interface have an address?

  ```
  /ip address print
  ```

- Do you have an address on your laptop's connected interface?
- Can you ping the gateway?
- Can you ping www.google.com?

Problems!

- Subnets on Router A can reach the Internet
- Subnets on Router B can't reach the Internet
- Subnets on Router A can't reach or respond to those on Router B
- This is because Router A does not know about the subnets on Router B
- We need to help it out.

Simple but wearisome

- The simple solution is to add static routes on Router A – one for each subnet on Router B
- Means manually updating routes on Router A every time we add/remove subnets on Router B
- Doesn't scale well to lots of subnets
- Doesn't scale well to lots of routers
- Plenty of room for error!

Less simple, but more better

- A better solution is to run a routing protocol such as OSPF
- Scales well (many routers, many subnets)
- After setup, only need to add things in one place
- The rest of the network updates itself
- Handles multiple paths

Static routes

- On Router A, add a static route for your subnet, pointing to the link address of Router B. E.g., for 192.168.7.0/24:

```
/ip route
add dst-address=192.168.7.0/24 \
      gateway=192.168.100.2
```

(the “add” is all on one line)

Test your static routes

- With your static route configured on Router A, you should be able to:
  - Ping any subnet on Router A from Router B
  - Reach the Internet from Router B
  - Reach hosts in your subnet from Router A

- Test other people's routes too :-)

OSPF routing (1)

- Give each router a (different!) loopback address:

```
/interface bridge
   add name=loopback
/ip address
add address=192.168.199.1/32 \
   interface=loopback
```

(the “add” is all on one line)

OSPF routing (2)

- Set each router's OSPF “router ID” to the same value as the loopback address:

```
/routing ospf instance
set 0 router-id=192.168.199.1
```

- The router ID is needed
- The loopback address is optional
- Just convention that they match

OSPF routing (3)

- On Router A, add all connected networks (except the WAN) to Area 0, e.g.:

```
/routing ospf network
add area=backbone \
    network=192.168.100.0/30
```

(the "add" is all on one line)

OSPF routing (4)

- On Router B, add the link network to Area 0, i.e..:

```
/routing ospf network
add area=backbone \
    network=192.168.100.0/30
```

(the “add” is all on one line)

OSPF routing (5)

- On Router B, add your subnet to Area 0, e.g.:

```
/routing ospf network
add area=backbone \
    network=192.168.7.0/24
```

(the “add” is all on one line)

OSPF routing (6)

- Check the routes on both routers:

  `/ip route print detail`

- "`ADC`" - these are dynamic, connected
- "`A S`" - these are static
- "`ADo`" - these are dynamic, OSPF
- You can remove the statics now...

Fully routed?

- At this point:
  - Every subnet should be able to reach every other subnet
  - Every subnet should be able to reach the Internet
- In short
  - We have a network :-)

Questions?